

KYMENLAAKSON AMMATTIKORKEAKOULU

Viestintä / Digitaalinen Media

Ilja Levonen

ERIKOISTEHOSTEET MOBIILPELIIN

Opinnäytetyö 2014

# TIIVISTELMÄ

## KYMENLAAKSON AMMATTIKORKEAKOULU

### Viestintä

Levonen, Ilja	Erikoistehosteet Mobiilipeliin
Opinnäytetyö	46 sivua
Työn ohjaaja	Siitonen Marko, Luennoitsija
Toimeksiantaja	Blackland Games Oy
Maaliskuu 2014	
Avainsanat	erikoistehoste, partikkeli, sprite-animaatio, mobiili

Opinnäytetyön tavoitteena on toteuttaa erikoistehosteet Blackland Games mobiilipeliin työnimeltään Planet Defender. Tehosteet luodaan Unity-pelinkehitystyökalulla käyttäen moottorista löytyviä particle emittereitä. Tehosteiden suunnittelussa on huomioitava, että tehosteissa käytetään mahdollisimman vähän partikkeleja, sillä kyseessä on mobiililaitteille suunniteltu peli. Mobiilialustoissa ei ole yhtä paljon muistia ja suoritin ei ole yhtä tehokas kuin muissa alustoissa, joten partikkelien määrä tulee minimoida. Blackland Gamesin puolelta ohjaajana toimi art director Tero Kuparinen. En ole tehnyt erikoistehosteita ennen, mutta olen piirtänyt tehosteita digitaalisissa maalauksissa, joten työn aikana olen oppinut animoimaan digitaalisia piirustuksia ja tekemään niistä sprite sheettejä.

Työ keskittyy lähes kaikkiin Planet Defender -pelin maailmassa tapahtuviin erikoistehosteisiin, kuten aseiden ammuksiin ja alusten räjähdyksiin. Suurin osa tehosteista luodaan tekemällä joko sprite-animaatioita tai käyttämällä tekstuureita Unityn emittereissä. En keskity ohjelmoitaviin tehosteisiin opinnäytetyössäni, sillä en käyttänyt niitä missään vaiheessa työharjoitteluani. Tutkin 2D-animaatioon ja erikoistehosteanimaatioon liittyviä työskentelytapoja sekä Unityn partikkeli-työkaluja.

Opinnäytetyön toteutuksessa käytetään Photoshop CS5 kuvankäsittelyohjelmaa ja Unity-pelimootoria sekä Blender ohjelmistoa 3D-mallinnusta vaativissa tehosteissa. Animaation luonti tapahtuu luomalla sprite-animaatioita Photoshopissa ja Unityn partikkeli-emitterissä.

## ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Media Communication

Levonen, Ilja

Special Effects for a Mobile Game

Bachelor's Thesis

46 pages

Supervisor

Marko Siitonen, Lecturer

Commissioned by

Blackland Games Oy

March 2014

Keywords

special effect, particle, sprite-animation, mobile

The objective of this thesis is to accomplish a set of special effects for mobile game by Blackland Games, and their project called Planet Defender. Effects are created with Unity game engine's particle emitters. While creating said effects we need to keep in mind that this game is designed to be played on mobile devices therefore particle amounts need to be kept to a minimum. Mobile platforms do not possess as much memory and processing power as many other platforms which means less particles to each given effect. Art director Tero Kuparinen was my supervisor while I worked at Blackland Games. I had not worked with special effects before this but I have painted effects in my digital paintings so during this project I have learned to animate and make sprite sheets out of them.

The project focuses on almost all special effects that take place in Planet Defender's game world such as ammunitions and explosions. Most of the effects are created by utilizing sprite sheet techniques or using painted textures in particle emitters. I have not been studying effects which need programming in this thesis since I have not used this technique during my practical training. Thus this thesis, focuses on Unity's particle tools and also on 2D and special effect animation.

In conclusion, this thesis has been carried out by using Photoshop CS5 photo manipulation program, an Unity game engine and a Blender for the effects that needed 3D objects. Sprite sheets have been animated in Photoshop and the special effects were animated in Unity.

# SISÄLLYS

## TIIVISTELMÄ

## ABSTRACT

1	SANASTO	5
2	JOHDANTO	6
2.1	Projektin lähtökohdat	6
2.2	Erikoistehosteista yleisesti	6
3	TEHOSTEEN LUOMINEN PELIIN	7
3.1	2D-animointi	7
3.1.1	Hahmoanimaatiosta elementteihin	8
3.1.2	Frame by frame	9
3.1.3	Elementtien animointi	11
3.2	Tekstuurikartta partikkeleille	13
3.3	Partikkelitehoste	18
4	PLANET DEFENDER	21
4.1	Työn aloitus	21
4.2	Planetary Guard -ammukset	21
4.3	Planetary Guard -panssarivaunu erikoistehosteet	29
4.4	Planetary Guard -puolustuslaitteet	34
4.5	Vihollistehosteet	37
4.6	Ympäristötehosteet	40
5	ITSEARVIOINTI PROJEKTISTA	43
	LÄHTEET	45

## 1 SANASTO

**Bake/beikata:** Pelien tekemisessä käytetty tekniikka, jossa luodaan esimerkiksi valmis kartta pelin valoista, joita ei tarvitse laskea reaaliajassa. Tekniikka vähentää rendaukseen kuluva laskentakapasitettia.

**Emitter:** Työkalu pelimoottorissa, joka luo partikkeleita.

**Frame:** Yksi kuva kuvasarjasta, joiden näyttäminen peräkkäin luo illuusion liikkuvasta kuvasta tai animaation.

**Mesh:** Kolmiulotteinen malli, joka muodostuu mesh-verkosta. Mesh-verkko rakentuu joukosta faceja, verteksejä ja edgejä. Face on mesh-verkon kolmikulmainen kaksikulotteinen pinta, verteksi on facen muodon kärkipisteet ja edge on useamman facen erottava linja.

**Partikkeli:** Kahdesta kolmiosta muodostuvia neliöitä, joille annetaan muoto tekstuurilla. Käytetään peleissä erikoistehosteiden luomiseen, esimerkiksi tulen simulointiin.

**Polygon:** Kaksiulotteinen tasokuvio suljetuilla reunoilla, jotka muodostavat 3D-malleja.

**Render/renderöidä:** Prosessi, jolla luodaan kuva tietokoneohjelmalla 3D-mallista. Renderöinnin rakenne dataan kuuluu rendattavassa ympäristössä oleva: geometria, kuvakulma, tekstuuri, valotus ja varjostus. Peleissä rendaus tapahtuu reaaliajassa tietokoneen näytönohjaimella.

**Shader:** Osa ohjelmointia pelimoottorissa, joka määrittää miltä tietty materiaali näyttää pelin sisällä 3D-mallissa.

**Sprite:** Kaksiulotteinen Bitmap-kuva tai -animaatio, joka on osa pelin visuaalisia elementtejä. Peleissä termiä käytetään mistä tahansa kaksikulotteisesta elementistä.

**Sprite sheet:** Kokoelma pieniä kuvia yhdessä suuressa kuvassa. Peleissä pelin moottori voi valita tietyn alueen suuresta kuvasta, jonka esittää ruudulla tai vaihtoehtoisesti näyttää useamman kuvan sarjassa.

## 2 JOHDANTO

### 2.1 Projektin lähtökohdat

Opinnäytetyön produktiivinen osa alkoi kesätyöni aikana. Olin jo opetellut perusteet Unityn partikkelityökalujen käytöstä ennen kuin sain varsinaisen opinnäytetyö-projektin. Blackland Games oli alun perin ottanut minut töihin osittain 2D-animaatio taitojeni takia. Työni ei kumminkaan kestänyt kauan, sillä alkukesän jälkeen projekti vaihtui toiseen projektiin. Syksyllä palasin työskentelemään Planet Defenderin tehosteiden pariin opinnäytetyö-projektin merkeissä.

### 2.2 Erikoistehosteista yleisesti

Erikoistehoste on illuusio tai temppu, joita käytetään elokuvissa, televisiossa, peleissä tai simulaatioissa, jotta voimme luoda erilaisia tapahtumia tarinassa tai virtuaalisessa ympäristössä. Erikoistehosteet jaetaan kolmeen osaan, optinen tehoste, käytännönläheinen tehoste ja tietokoneella luotu tehoste. Optinen tehoste luodaan yleisesti kameralla vaikka ylivalottamalla filmiä tai jälkityöstönä optisella tulostimella. (Film Reference 2014) Optista tehostetta voi käyttää näyttelijöiden asettamiseen eri ympäristöihin, joissa he eivät erikseen olleet. Käytännönläheisillä tehosteilla tarkoitetaan tehostetta, joka on oikeasti luotu kameran kuvattavaksi, kuten näyttelijän meikki, pyroteknikon luoma räjähdys tai suurella tuulettimella simuloitu tuuli (Westminster 2014). Tietokoneella luodut tehosteet ovat mitä nimi kertoo, tietokoneella luotu tehoste, vaikka matte-maalaus (maalattu tausta elokuvissa, millä luodaan illuusio ympäristöstä joka on liian kallis, mahdoton tai monimutkainen luoda todellisuudessa) tai räjähdys, joka on lisätty jälkeinpäin editoinnissa (Princeton University 2014).

Erikoistehosteet videopeleissä toimivat samalla periaatteella kuin edellä mainitut tehosteet. Kun peleissä puhutaan erikoistehosteista, niin monet käyttävät puhekielen

termiä "partikkeliefekti". Yleisesti monet ihmiset ajattelevat partikkelitehosteet uudenlaisiksi simuloinneiksi, joita tietokoneen tehokkaat grafiikka-kortit laskevat reaaliajassa. Todellisuudessa nämä tehosteet ovat lähempänä tulesta, savusta tai vaikka kipinöistä otettujen kuvien heittämistä kohti kameraa (Gamedev.net 2013).



**Kuva 1: Tulitehoste vuoden 1996 Crash Bandicootissa ja vuoden 2013 Tomb Raiderissa, joka on luotu samalla tekniikalla.**

Valtaosa peleistä käyttää vieläkin 2D kuvilla animoituja sprite-sheettejä tehosteissaan (Kuva 1). Nämä ovat hieman parempia verrattuna pelkillä partikkeleilla tehtyihin tehosteisiin. Varsinkin mobiililaitteilla, joiden suorittimien laskentateho ei riitä ylläpitämään useita partikkeleita ruudulla yhtä aikaisesti. Niin kauan kuin ruudulla on kohtuullinen määrä partikkeleita niin ongelmia ei ole, mutta pitää myös muistaa, että partikkeleita on vaikeampi hallita (InCreator[EST] 2014). Tämä voi johtaa siihen, että tehoste ei näytä uskottavalta. Erikoistehosteanimaattori Joseph Gilland sanoi, että suurin ongelma partikkelitehosteilla luodussa savussa on liiallinen symmetrisyys vaikkakin pienen turbulenssin lisääminen korjaisi tehostetta huomattavasti (Gilland 2009, 55). Sprite animaatio onkin käyttökelpoinen tapa luoda kohtuullisen realistisia tehosteita uhraamatta koneen laskentakapasiteettia.

### 3 TEHOSTEEN LUOMINEN PELIIN

#### 3.1 2D-animointi

Animaatio taiteen lajina on todella vanha. Yksi vanhimmista tunnetuista animaatioista ovat pilarit Egyptissä, joihin maalattujen hahmojen asennot muuttuvat hienovaraisesti jokaisen pilarin kohdalla. 1600 eaa. Jumala Isisille oli tehty 110 pilaria, joissa Jumala muuttaa asentoaan jokaisen pilarin kohdalla. Ensimmäinen pyrkimys heijastaa kuva

seinälle tapahtui vuonna 1640. Kyseessä oli Anthonasius Kircherin "Magic Lantern", jossa hän piirsi kuvia erillisille lasseille ja liikutti niitä valon edessä simuloidakseen liikettä. (Williams 2001, 12, 12-13.)

Kesti vuoteen 1896 asti ennen kuin saimme ensimmäisen filmille tallennetun animaation Thomas Edisonilta, joka oli kuvannut kamerallaan James Stuart Blacktonin piirroksia. 1900 luvulla perinteisemmin tehdyt animaatiot, kuten Felix Kissa ja Disneyn Mikki Hiiri, saapuivat valkokankaille. Television keksimisen jälkeen olemmekin saaneet paljon erilaisia animaatioita niin aikuisille kuin lapsillekin. (Williams 2011, 15, 17, 20.)

Pelit eivät ole mikään poikkeus tähän tapahtumaketjuun. Ainoa ero on tullut siinä mihin laitteisto on pystynyt aikoinaan. Pelit aloittivat yksikertaisilla pikselihahmoilla ja parin framin animaatioilla. Vuosien varrella pelikoneiden tehot paranivat, mikä vaikutti myös grafiikan ja animaation paranemiseen. Hyvä esimerkki on Playstation 2:lle vuonna 2006 julkaistu Metal Slug, joka käytti spritejä kaikkeen, hahmoihin, taustoihin ja tehosteisiin. 2D animaatiota hyödynnetään vielä tänäkin päivänä, etenkin erilaisissa mobiili -ja Indie-peleissä.

### 3.1.1 Hahmoanimaatiosta elementteihin

Erikoistehosteiden esi-isänä voidaan pitää George Méliès elokuvia (Gilland 2009, 3). Hän oli ensimmäisiä henkilöitä, joka alkoi tutkia tehosteita elokuvissa. Méliès käytti innovatiivisia matte-maalauksia, kaksinkertaista valotusta, ruudun jakoa, figureja sekä stop-motion animaatiota luodakseen ennennäkemättömiä kohtauksia. Nämä eivät olleet käsin piirrettyjä tehosteita, jotka tunnemme tänään, mutta niitä voidaan pitää lähtökohtana jokaiselle erikoistehosteelle tänään.

1900 vuosisadan alussa tehosteet olivat vain lisänä päähenkilön animaatioon (Gilland 2009, 5). Varjot, vesi, tuli, savu ja liike tehosteet luotiin vain osaksi hahmo animaatiota sen sijaan, että se olisi oma osa-alansa. Näitä olivat esimerkiksi: yllätyttämiseen liittyvä huutomerkki tai hikipisarot, sylkiessä syljen osuminen metalliväliin ja puuta hakatessa kirveellä pienten osien lentäminen ympäröivälle alueelle. Erikoistehosteilla ei välttämättä ollut omaa erikoistunutta animaattoriryhmää. Oli kyseessä mikä tahansa tehoste: aalto, nuotio tai tuulen vire puiden latvoissa, joku hahmoa työstävistä animaattoreista joutui tekemään sen.

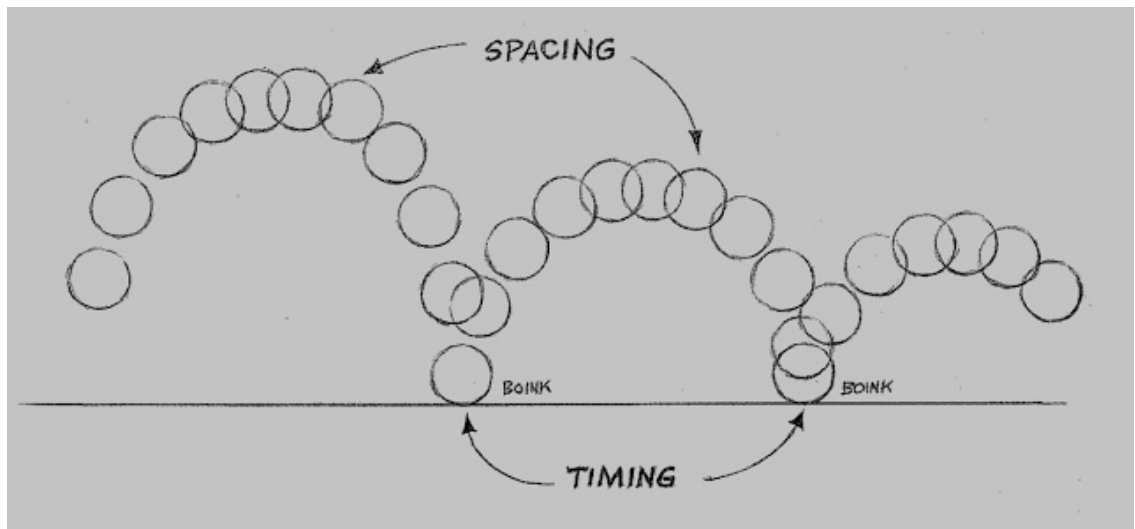


Winsor McCay:n animaatio "Gertie the Dinosaur" sisälsi innovatiivisen animaation veden animoinnissa, mutta jälleen kerran kyseessä oli enemmän hahmon esittely kuin itse veden tutkiminen animointia varten (Gilland 2009, 6). Kesti aina 1930 ja 1940 - luvuille, ennen kuin Disney toi erikoistehosteiden animoinnin uudelle ajalle. Tehosteiden animointiryhmään kuuluikin alussa vain kaksi ihmistä, Ugo D'Orsi ja Cy Young. He tulivat myöhemmin kuuluisiksi "Fantasia" elokuvan erikoistehosteanimoinnista. Fantasia-elokuvan takia animaatioryhmän kokoa kasvatettiin sadalla erikoistehosteisiin erikoistuneella taiteilijalla.

Tehosteiden animoinnista tuli toissijaisesta animaation osa-alueella yksi tärkeimmistä alueista animaatiossa luoden yhä todentuntuisempia tilanteita elokuvissa ja peleissä. Heidän ansiostaan meillä on uusi sukupolvi animaattoreita, jotka ovat kiinnostuneet luonnosta ja elementtien animoinnista. Se on johtanut suurempaan tutkimustyöhön, jolla varmistetaan, että jokainen aalto, nuotio tai puusta putoava lehti käyttäytyy realistisesti. Fantasia on elokuva, joka pohjautui eri luonnon voimien tutkimukselle, etenkin sen osio "The Rite of Spring", jossa on animoitu monia eri elementtejä.

### 3.1.2 Frame by frame

Peleissä oleva liikeanimaatio ei ole sen erilaisempaa kuin 3D-elokuvissa tai TV-piirretyissä. Suurin ero piilee pelikoneiden laskentatehoissa, jonka takia tarvitaan pienempiä polygon- ja partikkelimääriä. Kyseessä on pelien reaaliaikaisen renderaamisen vaatimukset vastaan animoidun elokuvan renderaamisen vaatimukset. Pelien grafiikan on toimittava reaaliajassa, koska pelaaja voi vaikuttaa pelimaailmaan. Elokuvassa taas esitetään valkokankaalla tapahtumia, joihin katsoja ei kykene vaikuttamaan, eli reaaliaikaisuutta ei vaadita. Elokuvissa tapahtuvien visuaalisten elementtien laskentaan voidaan käyttää useita satoja tietokoneita, kun taas peleissä käytetään tasan yhtä laitetta, joka joutuu laskemaan kaiken. Pelikoneiden laskentatehot ovat huomattavasti alhaisemmat. Silti tänä päivänä on helppo saada yhtä vakuuttavan näköisiä hahmoja, ympäristöjä ja tehosteita pelikoneiden tehojen kasvaessa.



**Kuva 2:** Richard Williamsin esimerkki ajoituksen ja framien määrän säätelystä (Williams 2011, 37).

Animaatioon vaikuttavat ohjeet eivät ole muuttuneet paljon vuosien varrella. Richard Williamsin ohjeet uskottavaan animaatioon ovat vieläkin päteviä. Hän on tunnettu "Who Framed Roger Rabbit" elokuvan animaatioista, joita pidetään tänäkin päivänä yhtenä parhaimmista esimerkeistä, kuinka sekoittaa 2D-animaatiota ja oikeita ihmisiä samaan tilaan. Hahmoanimaatiossa käytetyt ohjeet ajoituksesta ja frame määrän säätelystä voi suoraan verrata käsin animoitujen erikoistehosteiden piirtämiseen. Hyvä ajoitus ja framemäärä eri välivaiheille onkin tärkeintä animaatioita tehdessä (Kuva 2). Käytännöllisin tapa on rikkoa mikä tahansa liike ääriasentoihin ja pysäytyspisteisiin, joiden väliin tehdään tarvittava määrä välivaiheita (Williams 2011, 48-51). Riippuen siitä, pitäisikö liikkeen vaikka hidastua loppua kohden, teet loppuun enemmän frameja kuin alkuun.

Williamsin (2011, 61-63) mukaan animaatiotapoja on kolme: suoratapa, poseerauksesta poseeraukseen ja kahden edellä mainitun yhdistelmä. Suorassa tavassa animaattori piirtää ilman sen kummempaa suunnittelua ja katsoo miten animaatiolle käy. Tapa on todella spontaani ja yleensä johtaa huimiin oivalluksiin. Sen lisäksi animaattorille annetaan enemmän vapautta improvisoida ja muokata animaation kulkua. Ikävä kyllä se johtaa mitä todennäköisimmin mittakaavan ja pituuksien korjailuun, mikä tekee siitä pidemmän päälle kallista ja vaatii enemmän aikaa.

Poseerauksesta poseeraukseen animoidessa aloitetaan pääasenneista, joiden kuuluu olla animaatioissa, aina seuraavaksi tärkeimpään asentoon ja sitä seuraavaksi tärkeimpään asentoon ennen jokaisen asennon siirtymän suunnittelua. Tämä tapa antaa

selkeään suunnitelman animaation kulusta ja mahdollistaa sen, että toinen animaattori voi jatkaa sinun työsi loppuun, jos aikataulu tulee vastaan. Vaikka tämä tapa onkin nopeimpia tapoja animoida, se saattaa johtaa siihen, että animaatio näyttää lopussa luonnottomalta ja nykivältä. Animaation voi pyrkiä korjaamaan lisäämällä frame määrää, mutta animaatio ei pysy enää muodossaan. Tämä tarkoittaa esimerkiksi sitä, että animoidun hahmon raajat eivät pysy samassa koossa valtaosassa animaation aikana mistä seuraa hyytelömäinen liikehdintä.

Näiden kahden tavan yhdistely luokkin Williamsin (2011, 63) mukaan parhaimman tavan animoida ilman huonoja puolia. Tarkoitus on luoda animaation tärkeimmät ja toisiksi tärkeimmät kohdat ennen kuin hyppäät piirtämään jokaisen välivaiheen spontaanisti. Tarkoitus onkin tasapainottaa suunnitteleva ja spontaani puoli animoinnista ja tehdä siitä mielekkäämpää tekemistä. Williams (2011, 67) kertoo, että parasta on tehdä ensin tärkeimmät osa-alueet, sitten toisiksi tärkeimmät, kolmanneksi tärkeimmät jne. Lopussa lisätään kaikki liehuvat ja lepattavat osat kuten hiukset, läski, rinnat, hännät jne. Mikään näistä ei ole pakollinen sääntö animoinnissa, kyse on vain eri tavoista toteuttaa animaatioita.

### 3.1.3 Elementtien animointi

Erikoistehosteet ovat todella laaja käsite, joka vaatii siihen kuuluvan osaajaryhmän. Nämä animaatiot, liioiteltuja tai ei, vaativat suuren määrän osaajia ja aikaa jopa yksinkertaisimmassa tehosteessa. Jos et ole valmistautunut tarvittavalla määrällä tutkimustyötä, et tule pärjäämään ja hukkaat paljon aikaa. (Gilland 2009, 9.)

2D-animaation tarvetta kyseenalaistetaan entistä enemmän tietokoneiden aikakautena, sillä ne ovat helpottaneet elämäämme entisestään. Tietokone voi laskelmoida ja animoida nopeammin 3D:nä kuin animaattori, joka piirtää kuvasarjan 2D:nä. Silti 3D on velkaa 2D-ajan pioneereille. Kaikkein paras tapa alkaa tehosteanimointi on tutkia, piirtää, tuntea ja kokea elementti todellisuudessa vaikka tekemällä nuotio ennen kuin aloitat tulitehosteen tekemisen (Gilland 2009, 18). Kaiken A ja O on kyky tarkkailla ja imitoida. Tehosteita tehdessä on hyvä ymmärtää esimerkiksi miten ilmavirrat kuten kylmä ja lämmin ilma toimivat, mihin suuntaan ne kulkevat ja ennen kaikkea, mitä tapahtuu kun ne kohtaavat toisensa (Gilland 2009, 29).

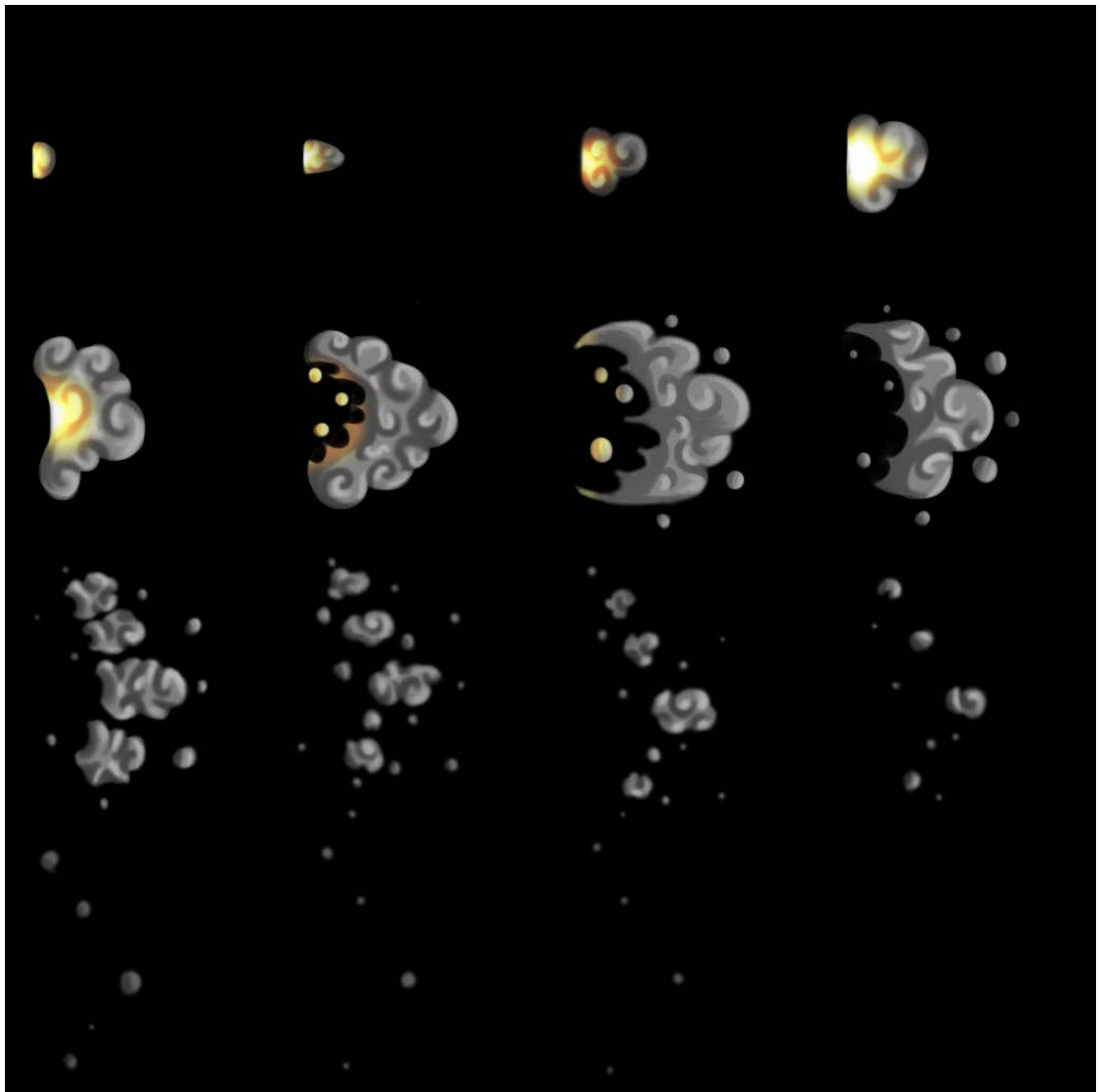
Tarkastellessamme vaikka savua, höyryä ja pölyä, huomaamme, että jokainen alkaa tismalleen samannäköisinä tehosteina. On huomioitava jokaisen elementin rakenne ja kuinka ne käyttäytyvät ilmassa. Savu liikkuu aina ylöspäin tulen tuottaman lämpimän ilman takia, jollei ilmavirtaus vaikuta siihen. Lisäksi savun haihtuminen vie eniten aikaa, joten se pysyy näkyvissä pidempään. Höyry käyttäytyy samalla tavalla ja kohoaa ylöspäin, mutta sen sisältämät vesimolekyylit imeytyvät nopeammin ilmakehään ja ympäristöön. Siitä syystä höyry haihtuu nopeasti ja muuttuu läpinäkyväksi. Pöly omaa raskaimman molekyylirakenteen näistä kolmesta ja vaatii tuulen tai räjähdysten noustakseen ilmaan. Se voi pysyä pitkiä aikoja ilmassa tuulen ansiosta, mutta ennemmin tai myöhemmin pöly laskeutuu takaisin maahan.

Gillandin mukaan tehosteen toteuttaminen kannattaa aloittaa yksinkertaisesta luonnoksesta, johon lisää yksityiskohtia myöhemmin. Vaikka poistat 80% yksityiskohdista pois se on silti vaikea animoida. Yksinkertaisella luonnosanimaatiolla on tarkoitus hakea ymmärrystä eri energioiden toiminnasta elementissä ja saada käsitys miltä animaatio tulee näyttämään (Gilland 2009, 48). Yksityiskohtia on aina helpompi tehdä lisää kuin poistaa. Kannattaa silti liioitella animaatiotaan, koska kyseessä on jotain, joka pyrkii vaikuttamaan todelliselta. Näin katsoja uskoo tehosteen todentuntuiseksi. Suurin osa aloittelevista animaattoreista ei liioittele animoimissa vaan kontrolloivat animaatiotaan tarpeettomasti. Tämä johtaa yleensä luonnottomuuteen ja tönkköyteen.

Kaikkia tehosteita tehdessä on hyvä muistaa pari ohjetta. Symmetrisiä linjoja tulisi välttää. Vaikka jotkin tehosteet olisivatkin symmetrisiä, olisi kuitenkin parempi liioitella ja tehdä tehosteesta epäsymmetrinen. Lopputulos näyttää paljon mielenkiintoisemmalta. Vältä myös yhdensuuntaisia linjoja. Luonnossa harvat linjat ovat yhdensuuntaisia, esimerkiksi savu, joka muuttaa linjojen paksuutta elinaikansa aikana. Tämä mahdollistaa realistisemman, mielenkiintoisemman ja dynaamisemman tehosteen.

Vesi on elementtinä tehosteanimaation kulmakivi, sillä sen dynamiikka näkyy kaikissa muissa elementeissä (Gilland 2012, 169). Monet vedessä näkyvät dynamiikat ja hahmotelmat voidaan käyttää edelleen esimerkiksi tulen animoinnissa. Esimerkkinä Disneyn "Bambie" elokuvan tuli on todella jäykän näköistä. Gilland (2012, 169) suosittelee, että tulta animoitaisiin kuin vettä.

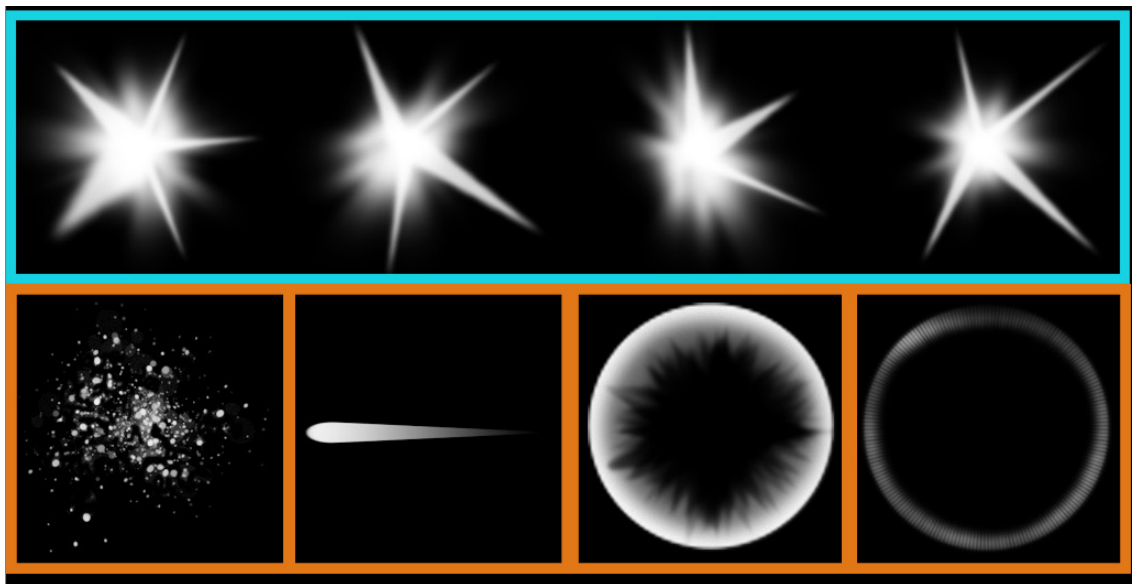
### 3.2 Tekstuurikartta partikkeleille



**Kuva 3: Sprite sheet.**

Spritejä käytetään moniin eri asioihin peleissä, räjähdyksestä lentäviin pirstaleisiin, kipinöihin, erilaisten ilmapiirien luontiin ja ilmiöihin, joita on liian hankala simuloida reaaliajassa (Crytek 2014). Lähes kaikki pelitehosteet alkavat sprite sheetin luonnista (Kuva 3). Sprite on kaksiulotteinen kuva tai video joka integroidaan tehosteeseen (Gamedev.net 2013). Sprite sheet on kokoelma eri spritejä yhdessä tekstuurikartassa, joita tehosteet käyttävät. Kartan sisältämät spritet voivat myös liikkua yhdestä toiseen luoden sprite-animaation. Kyseessä on silloin liikkuva tekstuuuri.

Koska yksi sprite sheet voi sisältää sekä animoituja että staattisia spritejä (Kuva 4), sen suunnittelu on tärkeää. Mikään ei estä tekemästä animoituja ja staattisia spritejä omiin karttoihinsa, mitä henkilökohtaisesti suosittelen. Ajatus sprite-animaatiossa on, että ruudulla näkyvä spriten ulkonäkö muuttuu tietyin väliajoin luoden illuusion liikkeestä, kun taas staattisen spriten ulkonäkö ei muutu vaan kaikki liike tapahtuu emitterin liikuttamisella. Mitä suurimmalla todennäköisyydellä tämä tapa korvataan tulevaisuudessa partikkelitehosteilla sitä mukaan kun ne paranevat vuosi toisensa perään (Ahearm 2009, 337). Tekniikka on kyllä kehittynyt tähän suuntaan, mutta ei ole kokonaan ohittanut vanhoja tapoja.

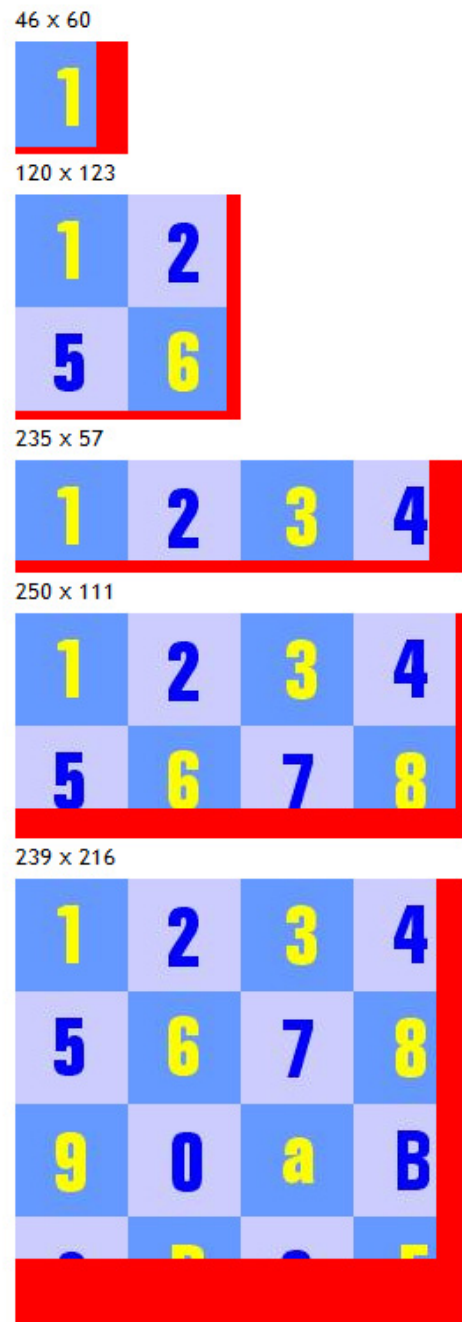


**Kuva 4: Esimerkki sprite sheetistä, jossa on animoituja ja staattisia elementtejä. Kuvassa sinisellä korostettu rivi on animoitu kuvasarja ja oranssilla korostutetut neliöt ovat staattisia elementtejä.**

These are **valid** or **recommended**



These are **invalid** or **not recommended**



**Kuva 5: Esimerkkejä Power of Two tekstuuri kartoista (KatsBits 2014).**

Sprite sheet on tekstuurikartta, joka on jaettu osiin. Ero muihin tekstuurikarttoihin on siinä, että jokainen jako on neliön muotoinen frame. Kuten normaalit tekstuurikartan, sprite sheettien on oltava "Power of Two" eli kahden potenssi säännön mukaisia (Kuva 5). Syy sääntöön on, että pelimoottorit on koodattu toimimaan niin, jotta pelimoottori voi prosessoida tietoa tehokkaammin (KatsBits 2014.)

Jokainen tekstuurikartta kannattaa luoda tämän mallin mukaan, sillä se on optimoiduin tapa peleissä. Säästät itseäsi turhalta korjailulta lopussa. Unityn partikkelityökalu osaa jakaa näin tehdyt tekstuurit omiin osioihinsa X ja Y koordinaattien avulla. Näin voit jakaa tekstuurin vaikka staattisiin paloihin tai animoituihin riveihin mahdollistaen useamman eri tehosteen spriten olemisen samassa sprite sheetissä.

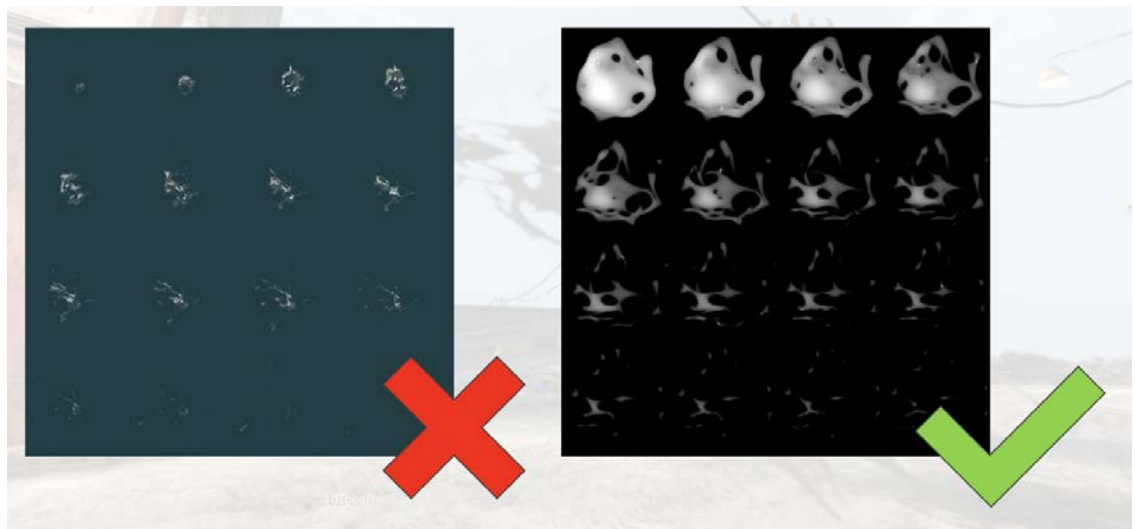
Tehosteiden kohdalla luot sprite sheetin kuin normaalin tekstuurikartan. Voit maalata sen digitaalisesti, käyttää valokuvia, tai vaikka luomalla 2D animaation ja jakamalla sen omiin frameihinsa ja asettelemalla ne paikoilleen sprite sheetissä. On kannattavaa huomioida seuraavat optimoinnin kannalta tärkeät yksityiskohdat.



**Kuva 6: Crytekin esimerkki tilankäytöstä sprite sheetissä (Crytek 2014).**

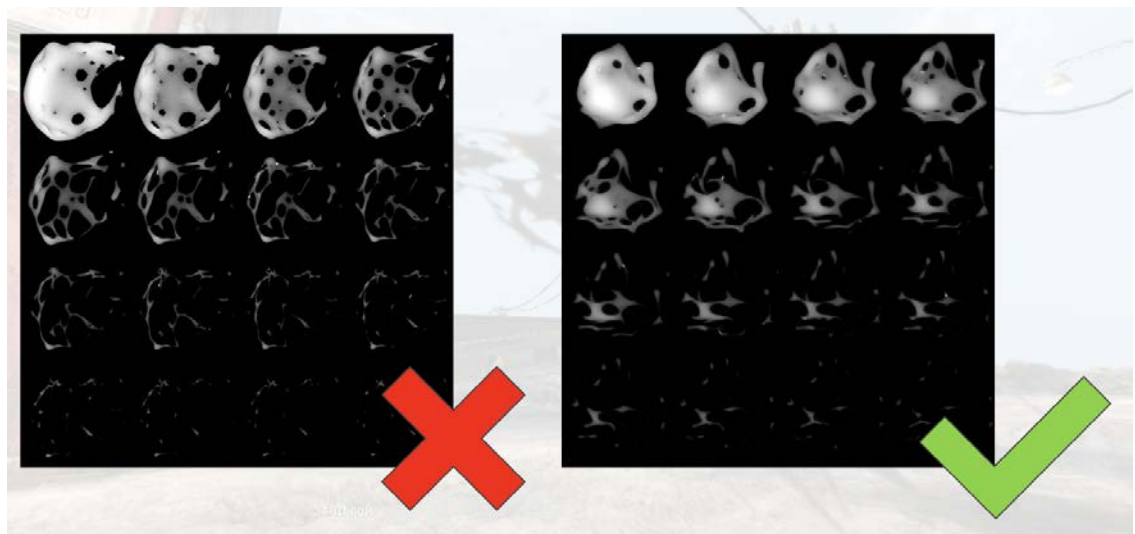
Älä tuhlaa tilaa (Kuva 6), on parempi hyödyntää vähäisinkin tila optimoinnin kannalta. Varsinkin jotta sprite sheetin kokoa voidaan pienentää hienosäädön aikana muistin säästämiseksi.





**Kuva 7:** Crytekin esimerkki monikäyttöisestä sprite sheetissä. Oikein tehtyä karttaa voi käyttää useammassa eri partikkelitehosteessa (Crytek 2014).

Pidä sprite sheet monikäyttöisenä (Kuva 7). Säästät prosessointia, jos pelissä tapahtuva tehosteet joutuvat lataamaan vain yhden sprite sheetin.



**Kuva 8:** Crytekin esimerkki yhdensuuntaisuuden välttämisestä sprite sheetissä (Crytek 2014).

Vältä yhdensuuntaisuutta. Väärin tehdyssä sprite sheetissä hajoaminen on yhdensuuntainen, kun oikein tehdyssä sprite sheetissä hajoaminen tapahtuu eri suuntiin. (Kuva 8). Joseph Gilland kertoi jo tästä, harvat asiat luonnossa ovat yhdensuuntaisia, joten saat paljon mielenkiintoisemman näköisen tehosteen.



**Kuva 9: Crytekin esimerkki dynaamisesta sprite sheetissä (Crytek 2014).**

Hyödynnä täyttä dynaamista vaihteluväliä (Kuva 9). Kuten aikaisemminkin, saat luotua paljon mielenkiintoisempaa visuaalista näyttävyyttä pelimaailmaan.

### 3.3 Partikkelitehoste

Partikkeli emittereillä luodaan pelin visuaalinen näyttävyys. Tehosteet ovat yksi asia, joita ei yleensä tuoda toisesta ohjelmasta pelimoottoriin vaan tehdään suoraan pelimoottorissa (Blackman 2013, 418). Tämä tapahtuu partikkeli-emitter komponentilla. Se luo joukon yksittäisiä partikkeleita ja antaa tehosteanimaattorin muokata yksittäisen partikkelin elinkaarta (Goldstone 2009, 187). Partikkeli itsessään on kaksiulotteinen kahdesta kolmiosta muodostettu neliö, jonka ulkonäköä muokataan tekstuurin ja läpinäkyvyys-kanavan avulla (Goldstone 2009, 187).

Unityssa on kaksi emitteriä, *Legacy* ja *Shuriken*. Näissä kahdessa emitterissä on pieniä eroja, kuten Shurikenin helpommat muokkaamismahdollisuudet emitterin tarjoamista vaihtoehtoista siinä missä *Legacy* mahdollistaa kokonaiseen muokkaukseen koodin kautta (Unity 3D 2014). Se kumpaa emittereistä käyttää on makuasia, koska kummatkin ovat pitkälti identtisiä, toinen on helpompi animaattorille ja toinen on helpompi koodaajalle. *Legacy* näyttää partikkelin geometrian editointitilassa, mikä voi haitata tehosteen tekemistä (Blackman 2013, 418). Emitterien lisäksi Unityssa on myös *Trail render* ja *Line render*. *Trail renderiä* käytetään aseista tuleviin ammusvanoihin, auton perävaloista muodostuviin valojanoihin ja muihin nauhamaisiin tehosteisiin. Tätä komponenttia ei tarvitse liittää partikkeliemitteriin,

vaan se toimii itsenäisenä komponenttina peliobjektissa. On kuitenkin suositeltavaa asettaa se tyhjään peliobjektiin ja asettaa tämä vanan omaavan peliobjektin alaiseksi hierarkiassa (Unity 3D 2014). Trail render on myös pysyvästi Billboard moodissa, eli se näkyy aina kameraa kohti, eikä tätä voi muuttaa. Line render on sama asia kuin Trail render, mutta staattinen, eli se ei liiku minkään perässä. Se luo suoran janan kahden tai useamman pisteen välille.

Partikkelitehoste, kuten muutkin objektit pelin sisällä, käyttävät shaderiä materiaaliinsa. Shader on ohjelmakoodi pelimoottorissa, jonka tarkoituksena on päivittää grafiikkaa reaaliajassa, esimerkiksi peilaavat shaderit voivat toimia reaaliajassa sen sijaan että ne olisi beikattu tai maalattu suoraan objektin pintaan (Ahearn 2009, 94-95). Erilaiset shaderit ovat hieman kuin Photoshopin blending modet. Esimerkiksi additiivinen shader muuttaa tekstuurin kirkkaammaksi ja partikkelitehosteessa se on sitä kirkkaampi mitä useampi partikkeli on päällekkäin (Kuparinen 2013). Toisin sanoen shader on matemaattinen kaava, joka määrittää miltä objektin varjostus tai materiaali näyttää. Tällä on suuri merkitys, miltä tehoste lopulta näyttää, joten oikean shaderin valitseminen oikeaan partikkeli emitterin materiaaliin on tärkeää partikkeli tehosteen luonnissa. Partikkelit voivat myös luoda ja vastaanottaa varjoja, mikä on tarpeellista, jos halutaan luoda esimerkiksi tulitehoste, jonka luomat varjot toimivat dynaamisesti pelimaailmassa (Goldstone 2009, 194).

Partikkeli-emitter siis monistaa niin monta partikkelia yksi toisensa jälkeen kuin tehostetta tekevä animaattori tarvitsee. Partikkelitehosteessa voit vaikuttaa yksittäisen partikkelin elinaikaan, kokoon, väriin, määrään ja nopeuteen (Goldstone 2009, 187). Partikkeli-emitterissä on myös oma partikkeli-rendaus, joka määrittää miltä partikkeli näyttää ja hallitsee tehosteeseen määritettyä materiaalia. Renderointi tapoja on viisi erilaista, Billboard, Stretched, Vertical Billboard, Horizontal Billboard ja Mesh Render (Unity - Game Engine 2014). Billboard kääntää partikkelit aina kameran suuntaisesti siinä missä Stretched kääntää ne liikesuunnan mukaisiksi. Vertical Billboard asettaa partikkelin litteäksi X/Z akselien mukaisesti kun taas Horizontal Billboard tekee saman X/Y akselien mukaisesti. Mesh Render käärii tehosteen materiaalin 3D-mallin ympärille luoden kolmiulotteisia partikkeleita. Yksi nokkelimmista tavoista luoda tulipalosta johtuva savu on käyttäen mesh renderiä. Koska savupartikkelit ovat raskaita mobiililaitteiden suorittimien tehon kannalta, on parempi tapa luoda mesh, jossa on animoitu savutekstuuri (ShadowGun 2013). Savu

liikkui animoimalla kahta tekstuuria päällekkäin ja meshin reunat pystyttiin pehmentämään vertex alphalla.

## Emitter hierarkia

### Räjähdys

↳ Savu

↳ **Välähdys**

**Pyöreä kipinä**

**Kipinä**

↳ Savujana

**Pirstaleet**

**Paine-aalto**



Kuva 10: Räjähdystehosteen emitter hierarkia.

Partikkelitehosteet voidaan jakaa kahteen kategoriaan, koko ajan toistuviin tehosteisiin ja hetkellisiin tehosteisiin (Blackman 2013, 421). Kokoajan toistuviin tehosteisiin voidaan luokitella esim. suihkulähteet, savupiipun savu ja osa valon lähteistä kuten nuotiot, joiden on tarkoitus pyöriä jatkuvalla toistolla pelimaailmassa. Hetkellisiin kuuluvat räjähdykset, ammukset ja muut tehosteet, jotka näkyvät ruudulla vain hetken ajan. Partikkelitehosteita luodaan suurimmaksi osaksi yhdistelemällä useampia partikkeli-emittereitä yhdeksi. Esimerkiksi partikkeliräjähdykseen voidaan käyttää useampia emittereitä, kuten tuli/savu, valon välähdys, kipinä, savujana (kipinöille), pyöreä kipinä, pirstale ja paineaalto (Gamedev.net 2013). Kun nämä emitterit laitetaan yhteen saadaan aikaan yksi räjähdys (Kuva 10). Jokainen emitter on päällekkäin paitsi savujana, joka on kiinni kipinä partikkelissa. Tietenkin kyseessä on vain yksi tapa luoda räjähdys, mutta jokainen räjähdys ei vaadi kaikkia kyseisistä emittereistä. Partikkelitehosteessa jokainen emitter on osa hierarkiaa, jonka "johtaja" määrittää elinajan ja "alaiset" toimivat perässä. Yksi suurimmista haasteista tehosteissa on saada emitterit synkronisoitua toimimaan oikeaan aikaan ja katoamaan heti sen jälkeen. Ajoitus on osa-alue, jota tehoste animaattori joutuu säättämään eniten saadakseen oikean näköisen tehosteen aikaiseksi.

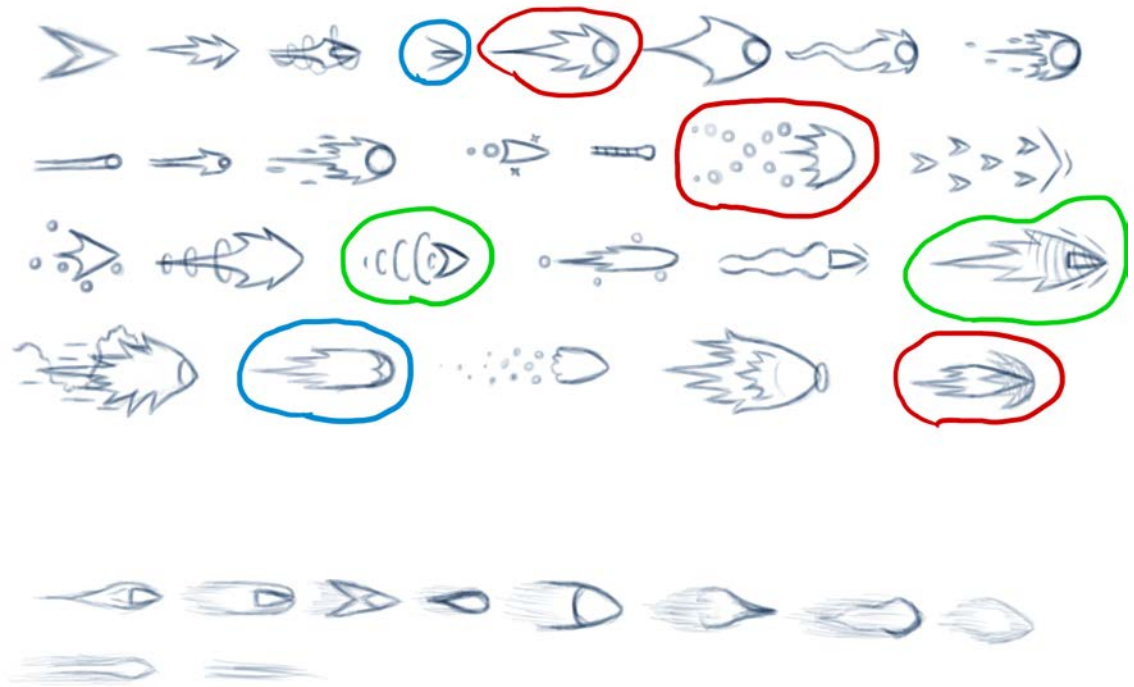
## 4 PLANET DEFENDER

### 4.1 Työn aloitus

Työn aloittaessani sain listan tarvittavista tehosteista, jossa oli jo joitain aiemmin tehtyjä tehosteita, mutta osa olisi tehtävä uudelleen tai muokattava. Lista oli aika mittava ja olin hieman epävarma listan pituudesta ja siihen liittyvästä työmäärästä, sekä projektiin käytettävästä ajasta. Aloitin kartoittamalla listaan, mikä oli jo valmista ja mikä ei. Osa tehosteista oli kuitenkin tehtävä uudelleen. Lista oli jaettu viiteen osaan tehosteiden käytön mukaan. Tarkoitukseni oli tehdä yksi alue kerrallaan.

### 4.2 Planetary Guard -ammukset

Blackland Gamesin Art director suositteli minua aloittamaan Planetary Guardin ammuksista, sillä ne vaativat vähiten uuden opettelua, sillä ne luotiin sprite-animaatioina. Pelissä olevista aseita oli aluksi neljä kappaletta. Jokaisella niistä on viisi päivitystasoa, joten jouduin suunnittelemaan myös miten panokset muuttuvat, jotta pelaaja tuntee olonsa palkituksi ja ase vaikuttaisi voimakkaammalta. Olin tästä syystä miettinyt tekeväni tehosteet osittain sprite animaatioina ja toisen osan emittereinä. Art director Tero Kuparinen ajatteli pidemmän päälle, että kun panoksia on ruudulla enemmän mobiililaitte ei jaksaisi pyörittää niitä, jos ne käyttäisivät useampia emittereitä. Oli paljon parempi käyttää mahdollisimman vähän laitteen suorituskykyä, jotta saisimme mahdollisimman monta vihollisalusta ruudulle.



**Kuva 11: Ammusten luonnoksia.**

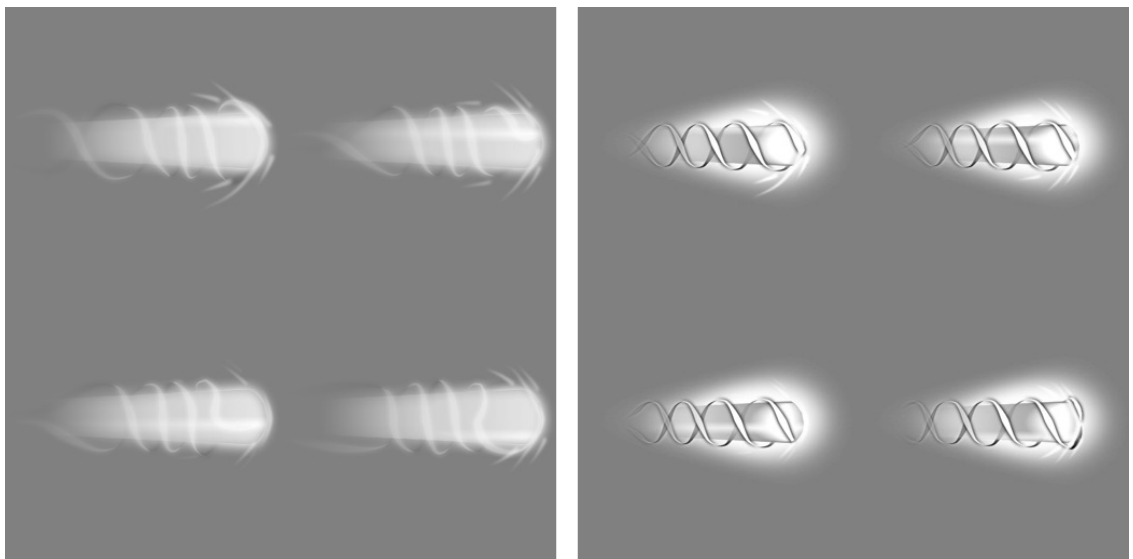
Ammusten suunnittelu alkoi paperilta, jolle luonnostelin erilaisia panoksia. Tutustuin eri shoot'em up -peleihin. Sen jälkeen tein useampia eri versioita (Kuva 11). Tutkimme aikaansaannokseni ja tulín siihen tulokseen, että voisin yhdistää elementtejä useammasta erinäköisestä luodista. Näin voisin luoda päivityksen keskitasolle erinäköisen luodin ja viimeiselle päivitykselle vain lisäelementeillä (Kuva 12). Kahdella ensimmäisellä ammustasolla ei olisi animoitua tekstuurikarttaa.



**Kuva 12: Päivitystasojen muutos alkuperäisen suunnitelman mukaan.**

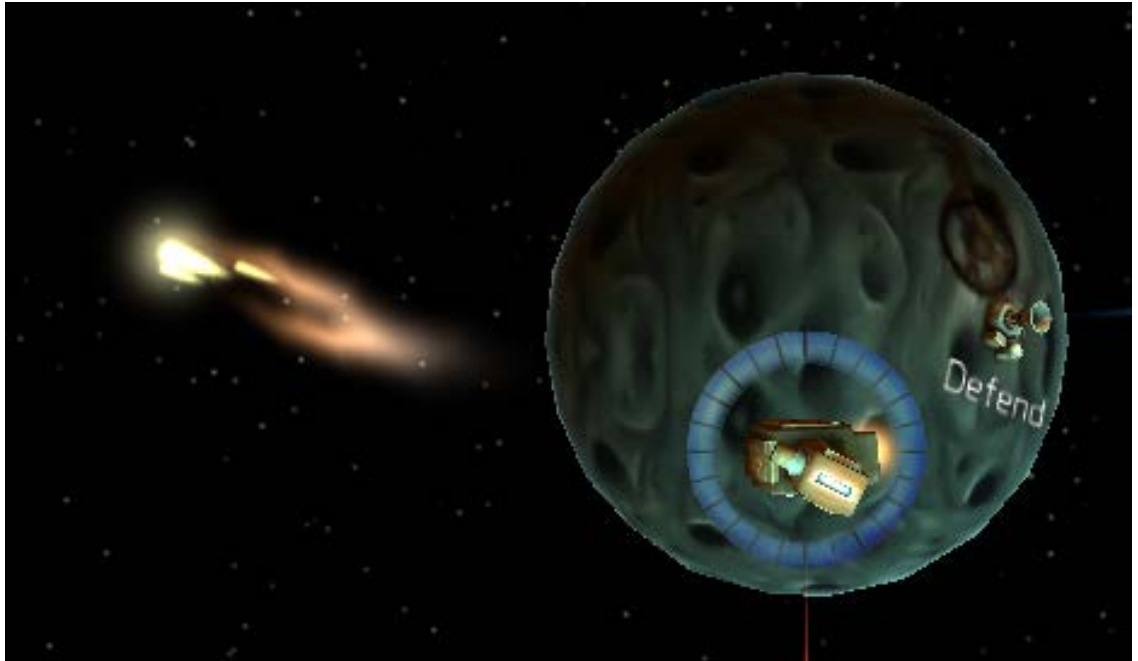
Aluksi ajattelin, että useampi tekstuurikartta rasittaisi muistia liikaa ja suunnittelin yhdistäväni tehosteiden tekstuurikarttoja, mutta Kuparinen kertoi minulle, miten pelin ohjelmointi toimii. Pelaaja valitsee vain yhden aseenn kerrallaan eikä voisi vaihtaa sitä kentän aikana. Tämä tarkoittaa, että usea pieni tekstuuuri olisi parempi vaihtoehto kuin yksi suuri tekstuuuri, jonka pelin pitäisi ladata joka kerta. Näin muistia kuluisi paljon vähemmän pelin aikana kun peli lataa vain aseeseen kuuluvan tekstuurin muististaan pelikentälle. (Kuparinen 2013).

Tehosteiden tekeminen alkoi Photoshopissa animoimalla jokainen frame erikseen. Sen lisäksi tein jokaisen sprite sheetin harmaan sävyisenä, jotta tehosteen värin pystyisi määrittämään emitterissä. Päädyin tekemään ammuksen neljän framen sprite-animaationa. Syynä tähän oli tiedosto ja tekstuuri koko. Ruudulla on jo paljon asioita kilpailemassa pelaajan huomiosta ja panokset eivät ole kovinkaan isoja, joten animaatio ei näkyisi paljon pelaajalle. On siis paljon parempi, että pelaaja voi mieluummin keskittyä tärkeämpiin elementteihin, mutta näkisi muutoksen. Ammuksen animaatioissa tarvittavien framien valmistuttua muutin animaation tekstuurikartaksi ja vein ne Unityn partikkeli-emitteriin. Aseteltuani emitterin arvot oikeiksi näin, miltä ammus näyttää pelissä.



**Kuva 13: Yksi ammuksista ennen ja jälkeen.**

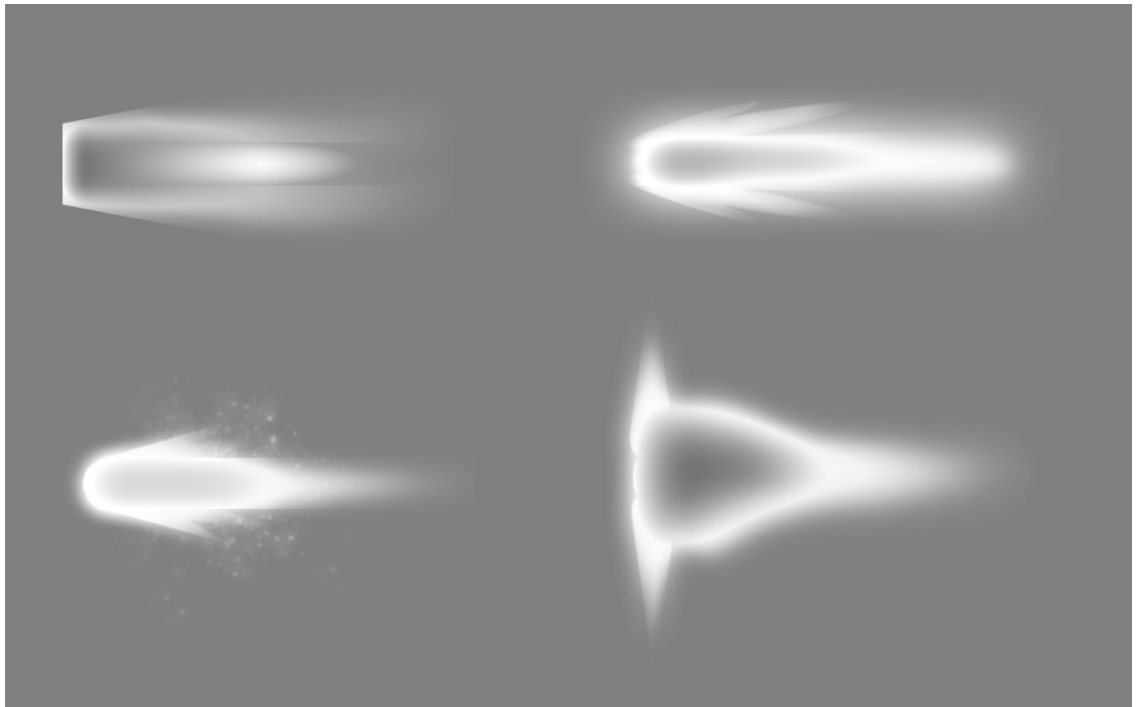
Sprite sheetiä tehdessä minua oli pyydetty tekemään tehosteista hienovaraisia. Planet Defender on visuaaliselta tyyliltään sarjakuvamainen, mutta jossain määrin myös realistinen peli. Jokaisella tehosteella on oltava oma sisäinen logiikkansa, joten hieman realistisempi kuvaus panoksista oli toivottavaa. Ikävä kyllä emme ottaneet huomioon, että hienovarainen tehoste ei toimi näin pienissä objekteissa, koska animaatio ei näkyisi pelaajalle lainkaan. Olin jo valmis antamaan työni mennä hukkaan tästä syystä, mutta en luovuttanut vielä. Toisin sanoen, jotta päivitystaso näkyisi selkeästi pelaajalle, sen täytyy olla sarjakuvamainen. Piirsin tehosteeseen selkeämmät linjat ja vähensin mahdollisimman paljon häivytyä (Kuva 13). Tehoste näkyi paljon paremmin, mutta se ei silti auttanut loppujen lopussa.



**Kuva 14:** Yksi lopullisista ammusmalleista pelinäköymässä.

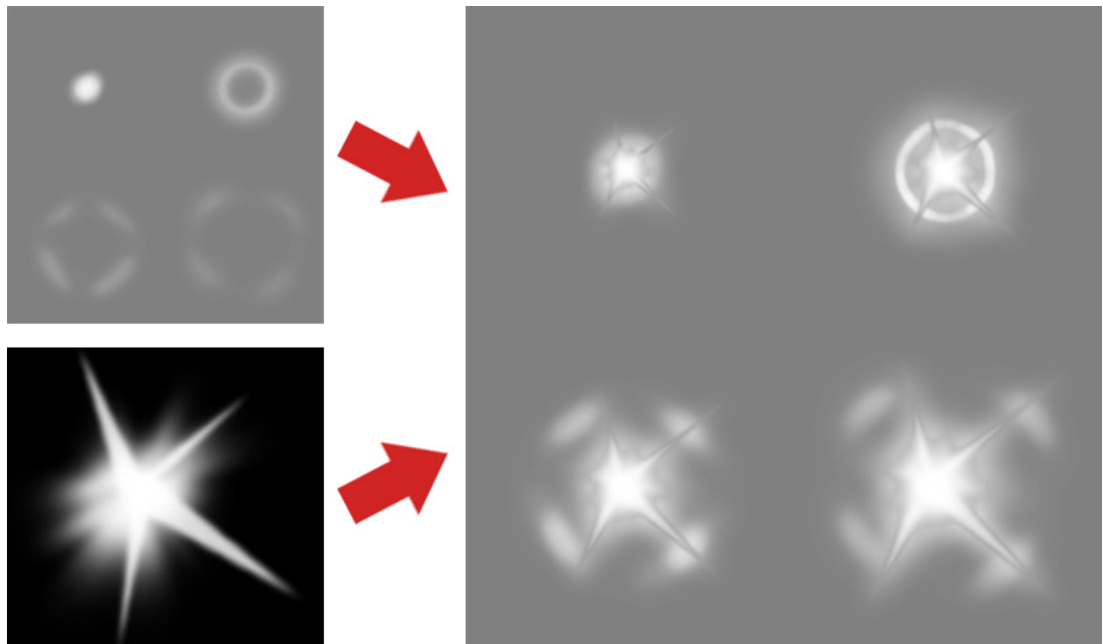
Testauksien jälkeen tulimme siihen tulokseen, että vaikka animaatio näkyi kohtuullisen hyvin, se ei ollut tarpeeksi. Jouduin toteuttamaan jokaisen aseeseen uudelleen, käyttäen useampaa partikkeli-emitteriä ja trail-rendereitä. Vaikka olimme alussa olettaneet sprite-animaation olevan parempi ajatus mobiililaitteiden suorituskyvyn kannalta, päädyimme käyttämään useampia emittereitä. Lisäksi minun oli luotava animaatio, jonka avulla trail render komponentti liikkui ammuksen luoneen emitterin mukana. Minun täytyi vain huomioida partikkelien määrän pitäminen mahdollisimman vähäisenä. Lopputuloksena oli kaikkein monipuolisin ja näyttävin ammus kokoelma, jonka tein parilla tekstuurilla (Kuva 14).





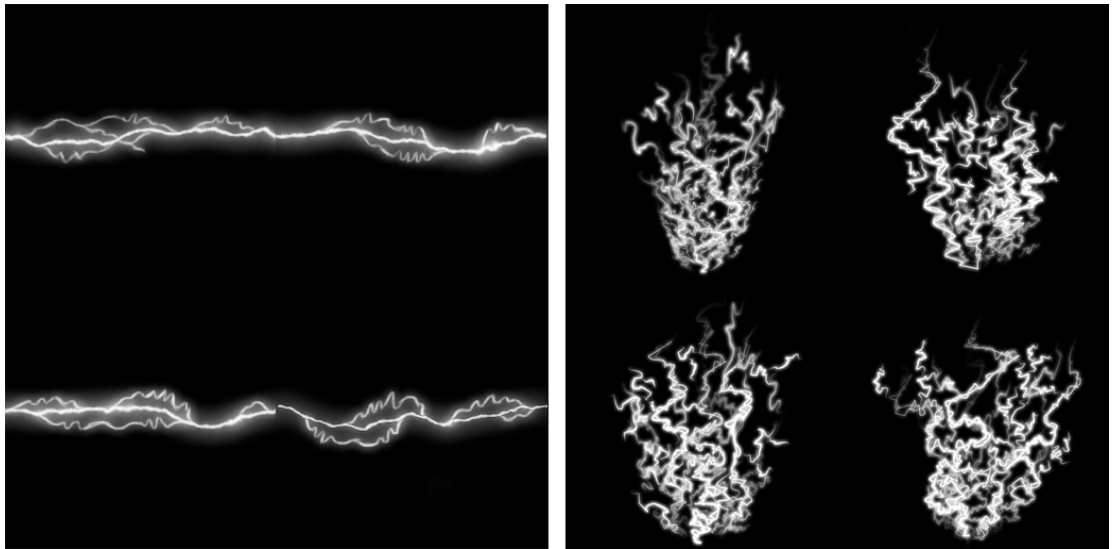
**Kuva 15: Aseiden suuliekkejä**

Aseiden suuliekit olivat todella helppoja luoda. Tein kymmenkunta erilaista suuliekkiä Internetistä löytyneiden mallikuvien pohjalta, joista valitsimme parhaimmat (Kuva 15). Suuliekki ei tarvitse animaatiota, joten työskentelyvauhti oli aikaisempaa nopeampaa. Testauksen tässä vaiheessa huomasimme ongelmaksi, että tehosteen pitäisi kääntyä 360 astetta jokaiseen suuntaan. Suuliekeille, sekä ammuksille tuli laittaa tietyt emitter asetukset, jotka hoitaisivat kääntämisen pelin aikana. Billboard render piti muuttaa Stretched Billboardiksi, joka kykeni kääntymään haluttuun suuntaan sen objektin mukaan, johon emitter oli kiinnitetty (Unity - Game Engine 2013). Suuliekeille piti tehdä vielä yksi muutos, jotta se toimisi moitteettomasti. Sille oli annettava skripti, joka tuhoaisi tehosteen pelikentältä heti kun se oli käytetty. Ilman tätä skriptiä kentälle jäisi suuria määriä näkymättömiä tehosteita, jotka ennemmin tai myöhemmin johtaisivat pelin nykimiseen mobiililaitteiden muistikapasiteetin vähyyden takia.



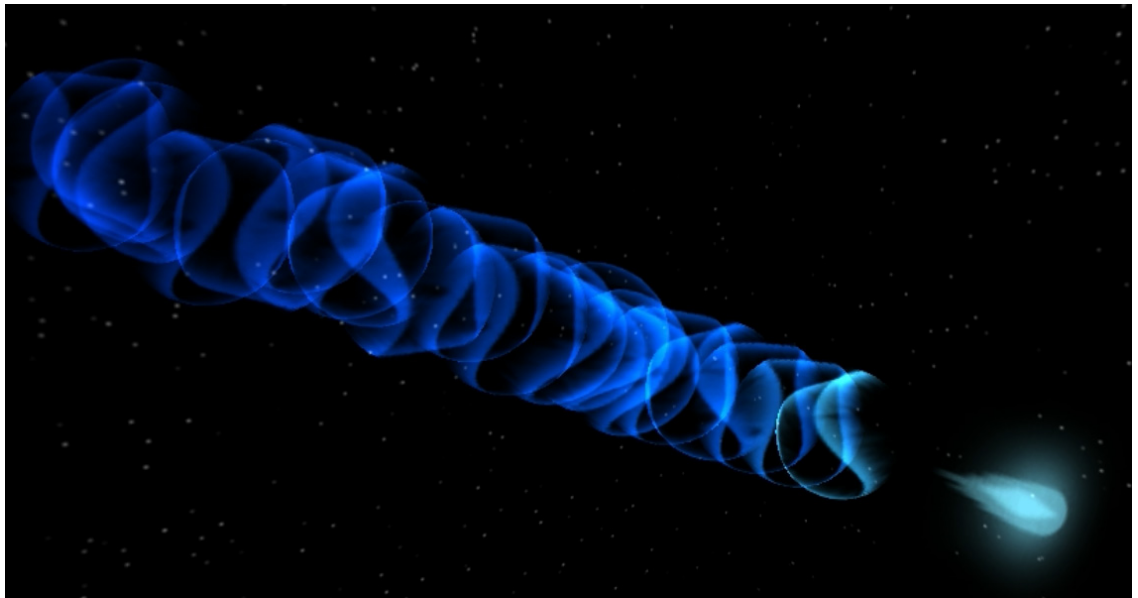
**Kuva 16: Esimerkki osumatehosteen sprite sheetistä ennen ja jälkeen.**

Ammuksiin liittyvät osumatehosteet olivat vaikeita luoda, koska en esimerkiksi tiedä miltä futuristisen laser säteen osuma näyttää kun se osuu alukseen. Normaali luodin hajoaminen on helppo selvittää Youtuben videoarkistoista. Toinen ongelma oli suunta, mistä osumaa tarkastellaan. Jos osumatehoste animoitaisiin sivulta, ongelma olisi sen kääntäminen oikeaan suuntaan ammuksen mukaisesti. Totesin, että se menisi turhaksi korjailuksi pidemmän päälle, joten animoin osuman leviämään joka suuntaan. Tein useamman vedoksen ennen kuin valitsin viisi suosikkiani. Unityssa kokeilun aikana Kuparinen pyysi minua yhdistämään tehosteita keskenään. Lopputuloksena loin viisi uutta tehostetta (Kuva 16). Tästä syntyi minulle omanlainen tapa työskennellä, sekoitin ja tutkin, mikä toimisi parhaiten ja missäkin tilanteessa. Loin tekstuureita yhdistämällä viisi uutta tekstuurikarttaa osumatehosteita varten. Näihin tehosteisiin lisättiin myöhemmin aluksen välkkyminen jokaisesta osumasta, minkä alus otti.



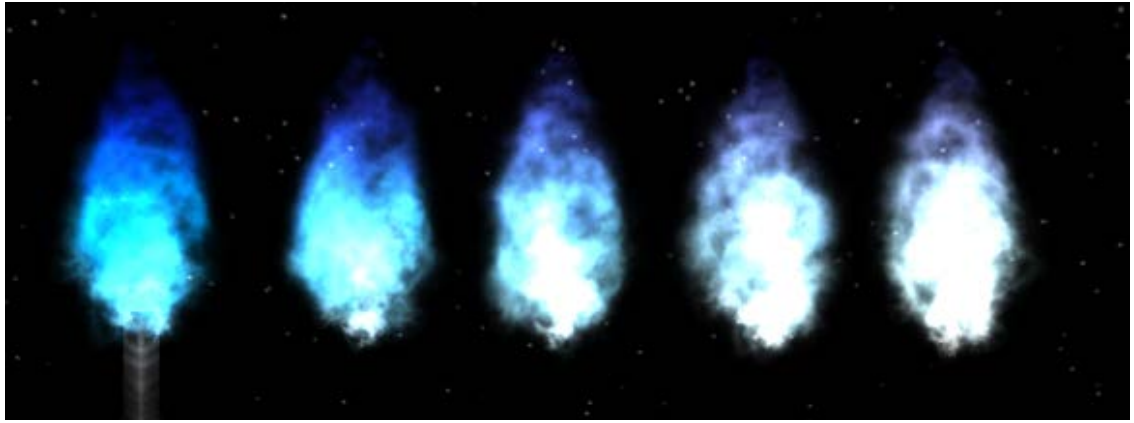
**Kuva 17: Tesla ammuksen ja suuliekin sprite sheet.**

Neljäs ase oli tuottanut minulle paljon ongelmia. Olin yrittänyt animoida sitä lähes viikon ilman tulosta. Päädyin laittamaan tämän tehosteen syrjään vähäksi aikaa. Kyseessä oli Tesla-salama ase, joka ampui salamoita vihollisia kohti automaattisesti näiden ollessa lähellä. Itse ammus (Kuva 17) ja osumatehoste olivat helppoja luoda, yksittäinen salama oli helposti animoitavissa neljään framiin. Osuma tehosteeseen kierrätin vanhaa tekstuuria kipinöiksi. Aseen "suuliekki" tuotti eniten ongelmia. Suurin osa ongelmista johtui siitä, miten animoin Tesla-salamaa. Alussa pyrin luomaan sen samalla tavalla kuin muutkin tehosteet. Jokainen frame jatkoi liikettä eteenpäin vaikka Teslan on tarkoitus olla täysin sattumanvaraisia sähköpurkauksia jotka osoittavat omiin suuntiinsa. Tämän ymmärrettyäni tein jokaisesta framesta niin erilaisen kuin pystyin. Lopullinen tehoste tehtiin kumminkin käyttäen ammustehostetta ja kopioimalla sitä tehosteessa "salamakimppua" varten. Loin salaman poikkeuksellisesti Photoshopin vapaan käden pen-tool kynällä, tapa joka oli minulle täysin uusi työkalu. Olin oppinut tämän tavan salama-animaatiota käsittelevässä videokurssissa ja se toimi hyvin tässä projektissa (Youtube 2014).



**Kuva 18: Railgun.**

Näiden aseiden jälkeen minulle annettiin pari ylimääräistä ammusta luotavaksi. Ensimmäinen oli Railgun (Kuva 18), joka oli tarkoitettu tarkka-ammunta kiväärin kaltaiseksi aseeksi. Minulla oli viides ylimääräinen ammusanimaatio jäljellä, jota ajattelin alustavasti käyttää lähtökohtana. Ajateltuani ammuksen nopeutta, tulin siihen tulokseen, että ammus on pelissä niin nopea, että animaatio ei näkyisi lainkaan. Sen lisäksi Railgunin latausaika on hidas, joten kentällä olisi vain yksi ammus nähtävissä kerrallaan. Aloin miettimään ammusta uudelleen. Otin inspiraatiota Arnold Schwarzeneggerin elokuvasta *Eraser*, jossa oli Railgun niminen ase, joka jätti jälkeensä savuvanan. Kuparinen piti ajatuksesta ja ehdotti minulle paineaaltorenkaan uusiokäyttöä tätä varten. Loin animoimattoman luodin, joka jätti jälkeensä rengasjanan, joka antoi vaikutuksen siitä, että luoti liikkui niin nopeasti, että se jätti paineaaltoa jälkeensä. Osumatehosteena käytin viidettä ylimääräistä tehostetta, jonka olin jo aikaisemmin tehnyt. Suuliekin tein normaaliin tapaan mallikuvista.

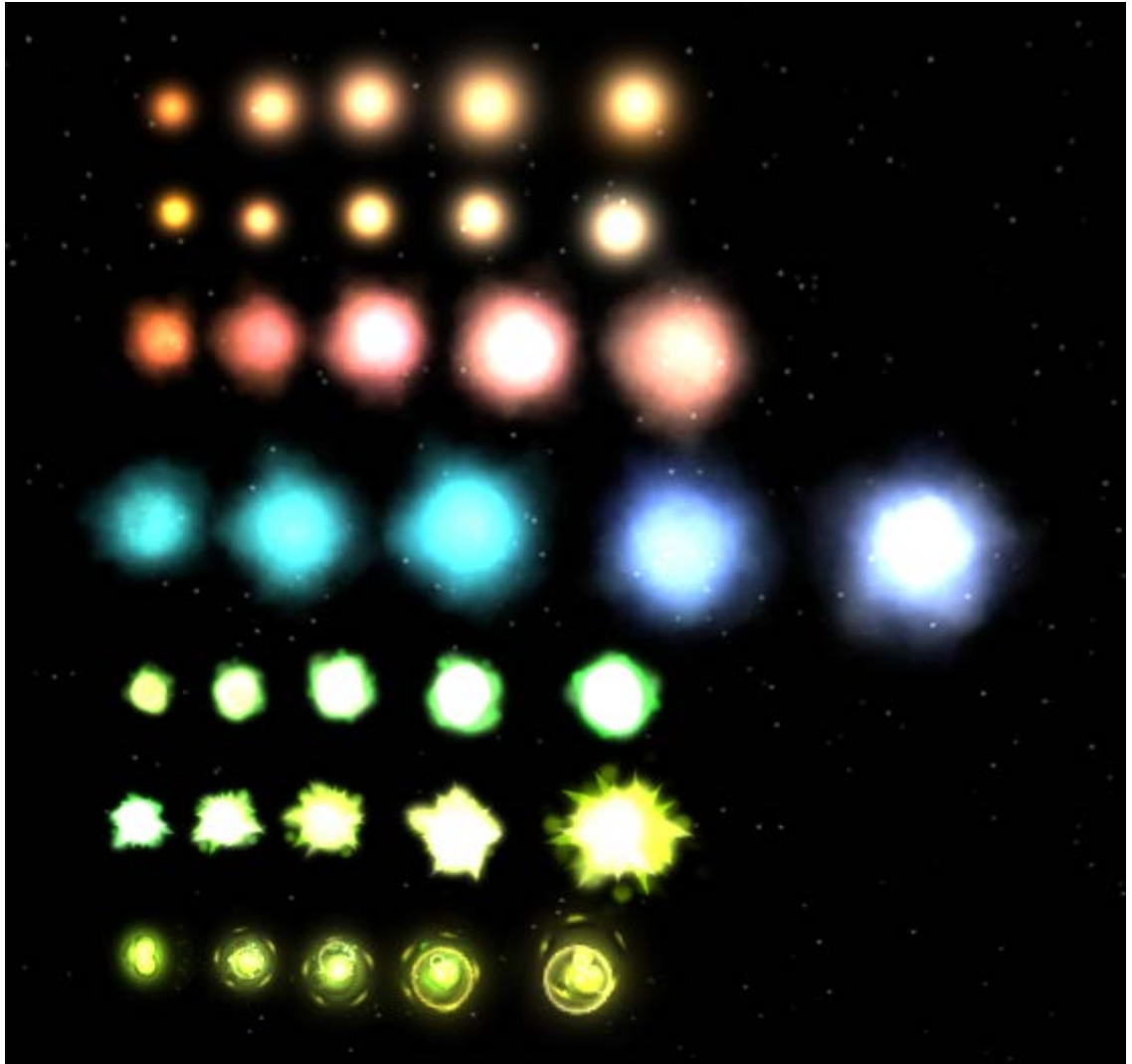


**Kuva 19: Raketin liekin viisi päivitystasoa.**

Jäljellä oli enää raketinheitin, mikä oli helpoin ammus luoda, sillä minun piti 3D-mallintaa ammus ja tehdä sille vain liekki. Kuten edellisissäkin tehosteissa kierrätin valmiita tekstuureja luodakseni pari uutta tehostetta todella nopeasti. Liekitechoste oli yksi tehosteista, josta pidin todella paljon. En ollut alussa varma, minkälainen tekstuuri liekille pitäisi tehdä. Tutkittuani liekin piirtämistä löysin tavan tehdä hyvännäköisen tekstuurin (Kuva 19). Käyttämällä Photoshopin filttareitä pystyin luomaan tarvittavan tehostetekstuurin tutoriaalin pohjalta (Photoshop tutorials for webdesigners 2014).

#### 4.3 Planetary Guard -panssarivaunu erikoistehosteet

Ammuksiin liittyneiden tehosteiden jälkeen jatkoin tekemällä panssarivaunun moottorin polttoliekin, niihin kuuluvat räjähdykset ja respawn tehosteen. Polttoliekkejä oli seitsemän kappaletta, joilla jokaisella oli viisi päivitystasoa. Olin hieman hätääntynyt ennen kuin ymmärsin, että näissä päivityksissä oli tarkoitus muuttaa lähinnä tekstuurikokoa ja väriä. Olin myös ajatellut tekeväni pari eri muotoista tekstuuria tehosteelle. Polttoliekki luotiin pallotekstuurista, joka monisti itseään loputtomiin. Koon muuttaminen tehosteen elinkaaren aikana loisi liekin muotoisen tehosteen. Kuparinen oli myös ehdottanut, että uusien tekstuurien luomisen sijaan voisin käyttää vanhoja luodakseni uusia tehosteita. Tein juuri näin.

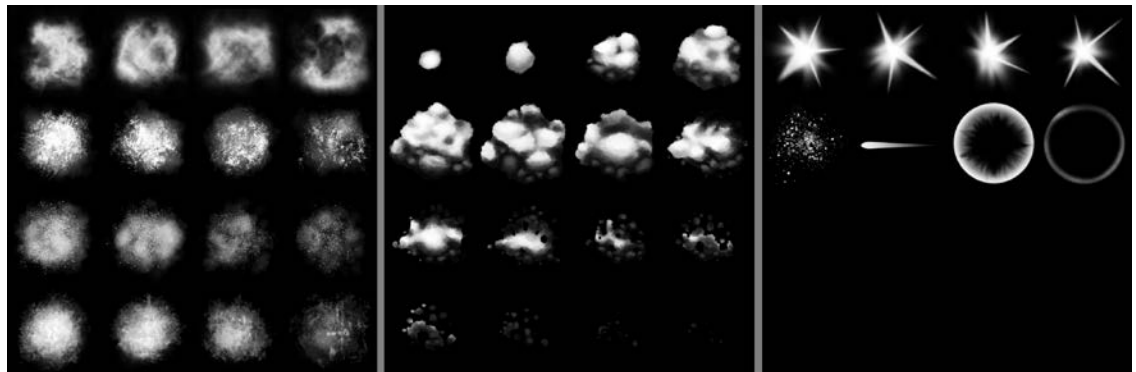


**Kuva 20: Aluksen päästöliekit.**

Lopputuloksena oli toistakymmentä tehostetta (Kuva 20), jotka hyödynsivät vanhoja materiaaleja. Kaikkein parhaiten toimivat osumatehosteet ja räjähdysiin käytettävät savutekstuurit. Suuliekeistä sai todella komeita liekkejä, mutta niissä oli ongelma, kääntyminen. Joka kerta kun alus kääntyisi, liekki ei seuraisi kunnolla perässä tai edes oikeinpäin. Stretched billboard korjasi osan ongelmista, mutta kuten nimi sanoo, se venytti tekstuuria, joten nyt sijoitus tuli ongelmaksi, puhumattakaan siitä kuinka tehoste meni aluksen läpi vähän väliä. Pitkin kynsin jouduin päästämään irti tästä tehosteesta. Pidättäydyin ympyrän muotoisissa tekstuureissa ja loin tarvittavat polttoliekit. Päivityksen oli tarkoitus liikkua selvästi puhtaampaan liekkiin likaisesta. Näin pelaaja näkisi, että moottori olisi selvästi kehittynyt eteenpäin.

Ydinreaktorimoottoreiden liekeissä sain leikitellä eniten, sillä niillä ei ole selvää esikuvaa, josta itse tietäisin. Ainoa ohje oli, että niiden tulisi näyttää epävakailta ja

vaarallisilta. Käytin niissä visuaalisesti mielenkiintoisimpia tekstuureja. Sen lisäksi laitoin liekin heilumaan edestakaisin nopeasti. Näin liekki näyttää oudolta, epävakaalta ja vaaralliselta.



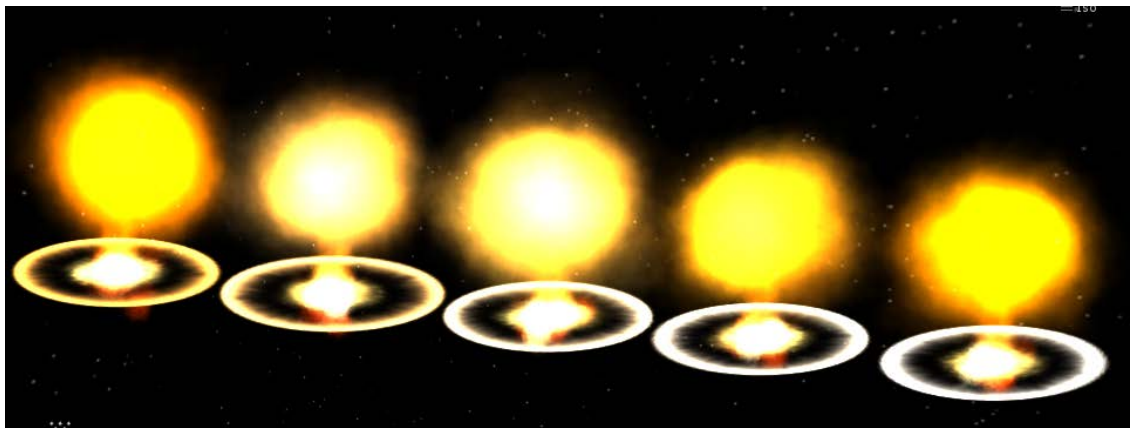
**Kuva 21: Räjähdyksissä käytetyt sprite sheetit.**

Jokainen moottori toi myös mukanaan oman räjähdyksensä, kun pelaaja kuolee. Tein yhden 16:sta framen savupölylähdyks sprite sheetin. Sen lisäksi tein neljästä neljän framin animaatiosta koostuvan tekstuurikartan, jolla voin luoda enemmän erinäköisiä räjähdysä helpommin. Loin myös tekstuurikartan räjähdysäin kuuluvia lisäelementtejä kuten paineaalto, tuli ja kipinät (Kuva 21). Sen lisäksi otin yhden vanhan savu tekstuurin toisesta projektista, johon se ei ollut kelvannut. Jokainen teksturi oli luotu maalaamalla Photoshopissa hyödyntäen erikoissiveltimiä. Näillä tekstuureilla pystyttiin luomaan helposti seitsemän erilaista räjähdystä moottoreille käyttäen eri teksturiyhdistelmiä (Kuva 22). Päivitys tehtiin samalla tavalla kuin polttoliekeihin muuntamalla väriä ja kokoa hienovaraisesti. Suurimmat muutokset tapahtuivat kolmannen ja viidennen taso kohdalla.



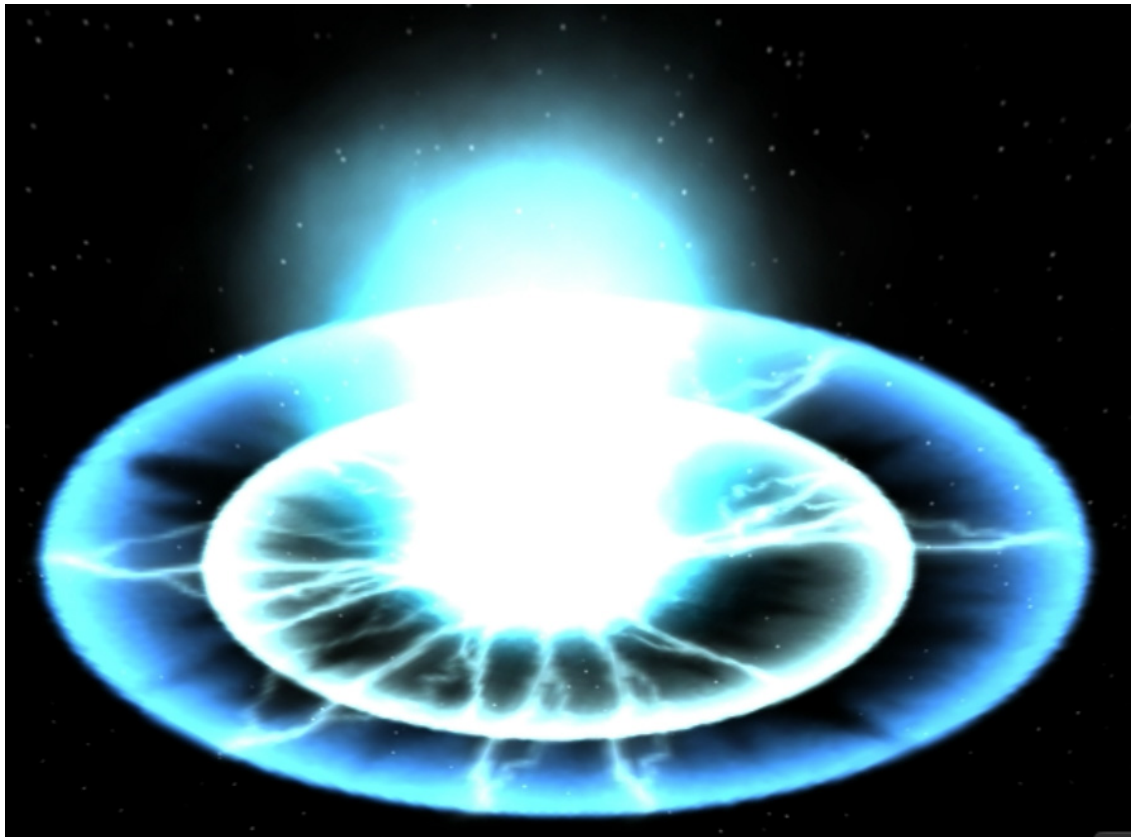
**Kuva 22: Moottoreiden räjähdysä.**

Ydinräjähdykset olivat kuitenkin paljon rajatumpia ja selkeitä. Jokainen meistä tietää, miltä räjähdys tulisi näyttää, joten haasteeni oli luoda oikean muotoinen räjähdys. Koska pelaaja ei näkisi räjähdystä muualta kuin yläpuolelta olisin voinut tehdä tehosteen helposti ilman sienimuotoa. Pysin kuitenkin tekemään sienimuodon (Kuva 23) kierrätyskelpoisuuden vuoksi. En voi olla varma siitä, käytettäisiinkö kyseisiä räjähdyskuvia tai variaatioita niistä muualla pelissä niin, että pelaaja näkisi räjähdyskuvan vaikka sivusta. Oli parempi nähdä enemmän vaivaa nyt sen sijaan, että joutuisin tekemään sen myöhemmin. Kulutin useamman tunnin muodon kanssa. Emitterit eivät toimineet muodon luomisen suhteen, joten joudun luomaan 3D-mallin auttamaan muodon kanssa. Loin ylösalaisen suppilon jalaksi ja hatun "sienelle" Blenderin 3D työkaluilla. Hattu auttoi muodon saavuttamisessa, mutta jalka oli loppujen lopuksi turha, sillä sain sen luotua parilla emitterillä paremmin. Siitä huolimatta opin hyväksikäyttämään 3D malleja tehosteiden luonnissa.



**Kuva 23:** Ydinräjähdyskokeilu.



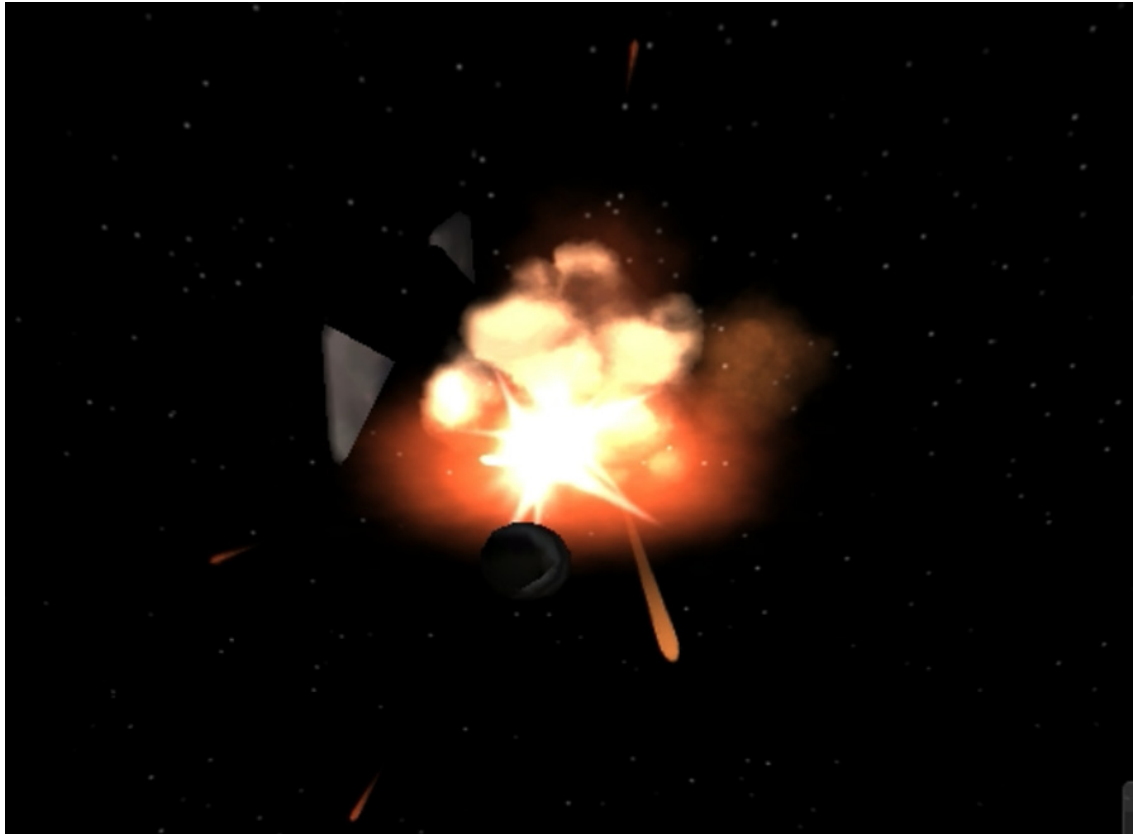


**Kuva 24: Respawn tehoste.**

Respawn tehoste oli jotain, mistä minulla oli oma näkemys siitä miltä efektin tulisi näyttää, kun pelaaja saapuu takaisin kentälle kuoleman jälkeen. Kuparinen oli antanut pienimuotoista ohjeistusta tehosteelle. Tästä huolimatta päätin mennä omaan suuntaani. Jos Kuparinen ei tykkäisi siitä, tekisin sen uudelleen, aikataulu antaa periksi sen verran, että voisin kokeilla ajatustani. Ajatukseni oli luoda pulssiva sähkökenttä, jonka keskelle alus materialisoituisi välähdyksessä (Kuva 24). Loin tämän kierrättämällä räjähdystekstuuria ja Teslan sähköammusta. Kaksi paineaaltoa pyörisivät sisäkkäin vastakkaisiin suuntiin samalla kun niiden keskellä pyöri kuusi salamaa eri suuntiin. Pulssisykäyksiin käytin Kuparisen valmiiksi luomaa 3D puolipalloa, joka helpotti työn tekoa huomattavasti. Lopulta näytin tehosteen Kupariselle, ja hän piti siitä todella paljon. Hän kuitenkin antoi minulle parannusehdotuksen, jonka mukaan minun oli tarkoitus luoda valon säde, joka ampuisi itsensä planeetasta. Tällä oli tarkoitus selventää pelaajalle, että uusi alus lähetetään planeetalta. Tein ensimmäisen version emitterillä, joka toimi nätisti, mutta Kuparisen mielestä olisi parempi, jos tekisin tehosteen 3D-mallilla. Tein sylinterin Blenderillä ja animoin sen Unityn animaatiotyökalulla venymään ylös. Ongelma oli kumminkin materiaali ja jostain syystä malli ei toiminut materiaalin kanssa. Unityn peruskappaleista löytyvä sylinteri olisi toiminut, mutta pivot point oli väärässä

paikassa, joten animaatio ei olisi toiminut oikein. Lopuksi päädyin tekemään tekstuurin, joka kasvoi pienestä koosta isoksi. Tehostetta katsottaisiin aina ylhäältäpäin, joten se loisi illuusion säteestä, joka tulee planeetan pinnalta.

#### 4.4 Planetary Guard -puolustuslaitteet



**Kuva 25:** Tykkitornin tuhoutumisräjähdys.

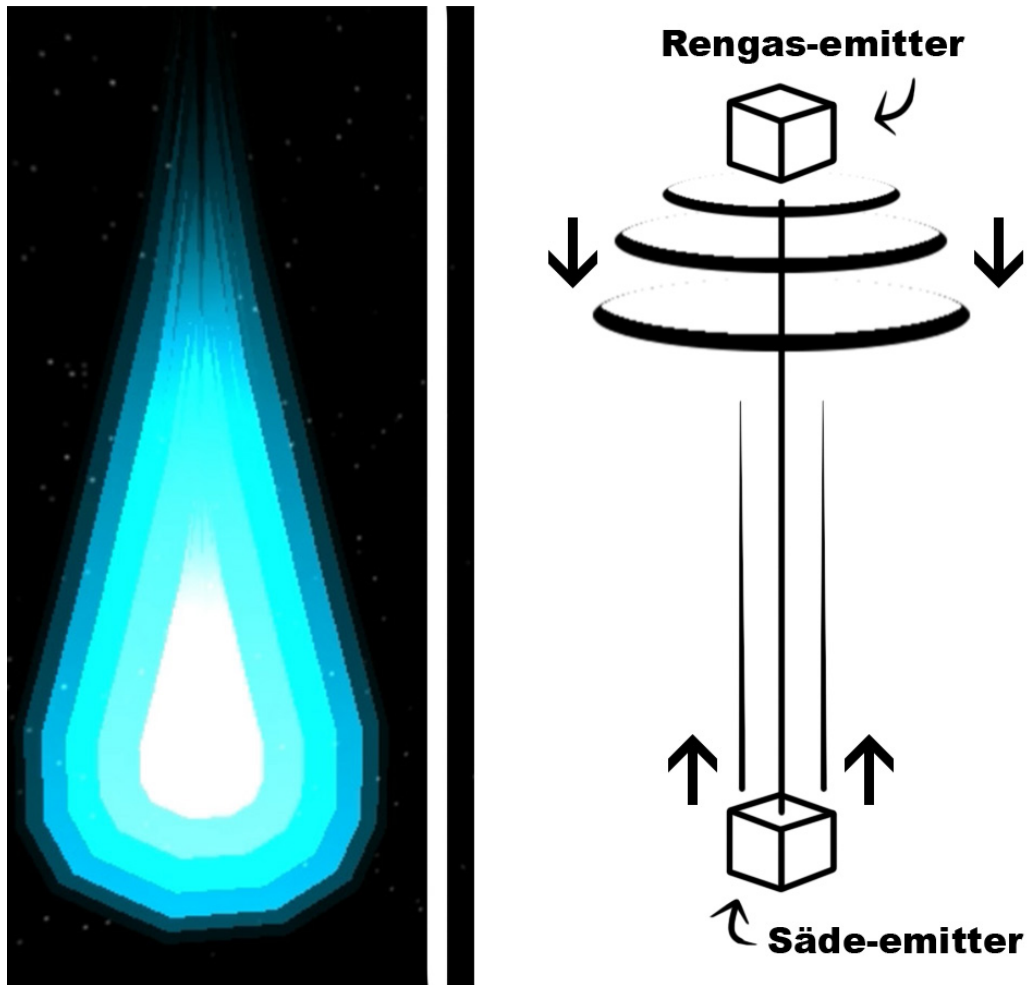
Viimeisenä Planetary Guardiin kuuluvista tehosteista olivat puolustukseen keskittyvien aseiden tehosteet. Suurin osa oli vain erilaisia räjähdyskuvia esimerkiksi ohjuksille, kranaateille ja miinoille, joten hyödynsin jo valmiiksi tehtyjä tekstuureja. Ainoa uusi asia, jonka tein parille kuolinvälikappaleelle uutena ominaisuutena oli sattumanvaraisten osien sinkoutuminen räjähdyksestä (Kuva 25). Tämä osuus tehtiin emitterillä, joka ampui valmiiksi tehtyjä ja teksturoituja 3D paloja. Palojen ongelma tehosteissa oli, että palat katosivat ilmaan. Tämä ongelma korjattiin sub emitterillä, joka aktivoitui 3D mallin animaation lopussa. Mallien kadotessa paikalle ilmestyy räjähdys, joka tuhoaa palat nättisti.



**Kuva 26: Hold the line.**

Toinen 3D-malleja hyödyntävä tehoste oli "Hold the Line" (Kuva 26), eräänlainen aluksen tehoja lisäävä toiminto, kun pelaajan terveys tippuu liian alas. Ajatuksena oli tehdä sykkivä aura aluksen ympärille. Tehoste oli helppo tehdä, kun emitter loi aluksen ympärille sinisenä sykkivän muodon. Tehosteesta pidettiin, mutta ongelmaksi tuli aluksen päivityksessä tapahtuva aluksen 3D-mallin muuttuminen. Toisin sanoen mallin muoto muuttuisi ja auran muodon tulisi myös muuttua. Tehostetta päädyttiin käyttämään ilman alusmallia hologrammina, joka vetää vihollistulen itseensä.

Pelaajalla oli toissijaisia aseita kuten energia-aalto, jolla vihollisia pystyisi työntämään kauemmas. Tämä oli helpoin tehoste tehdä, sillä se oli vain paine-aalto, jota käytettiin jo räjähdyksissä. Ainoa ero oli, että aseensa luoma paine-aalto on sininen. Vetosäde oli paljon mielenkiintoisempi, sillä jouduin oikeasti miettimään miten saan luotua vaikutuksen siitä, että pelaaja vetää asioita itseensä päin. Jokainen emitter puskee tekstuuria pois päin itsestään enkä tiennyt, miten emitterin saisi toimimaan päinvastoin. Tein perussäteen, joka puski renkaita ulospäin, mutta halusin renkaiden toimivan toiseen suuntaan jollain tavalla. Helpoin tapa saada tehoste toimimaan näin oli kääntää rengas emitter väärinpäin säteen päähän. Nyt säde lähti eteenpäin ja renkaat lensivät takaisin sinne mistä säde tuli (Kuva 26).



**Kuva 27: Orbital strike ja vetosäteen rakenne**

Minulle annettiin lopuksi vielä pari tehostetta luotavaksi viimeisillä viikoilla.

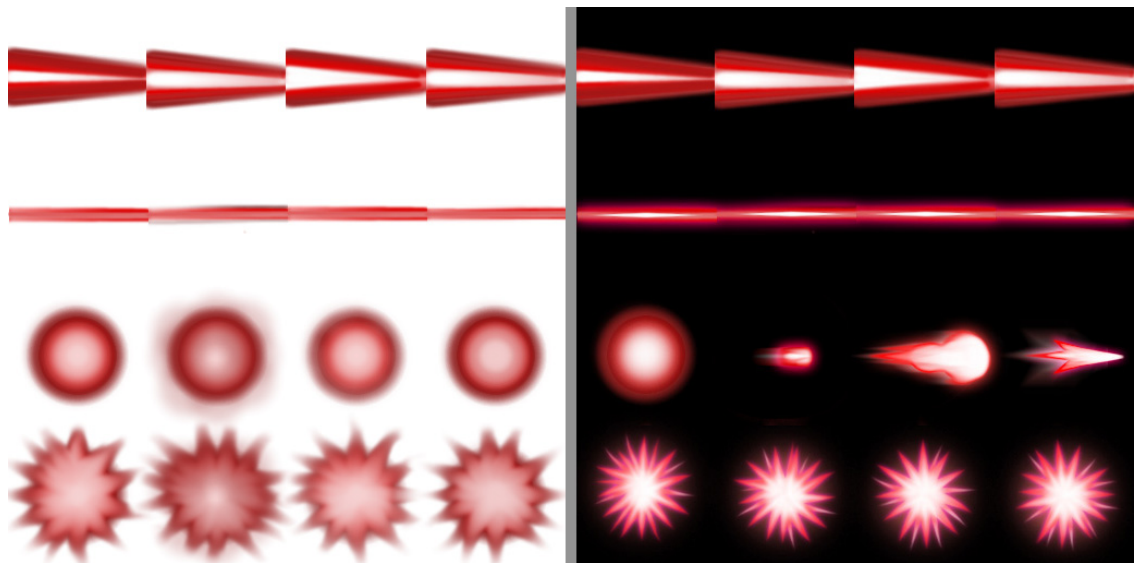
Kuparinen halusi minun luovan ilmahyökkäyksen, joka tyhjentäisi kentän vastustajista. Ajatukseni oli luoda laser, joka ampuisi itsensä taivaalta ja loisi suuren paineaallon osumasta. Kuparinen halusi minun kuitenkin luovan vesipisaran, joka vaikuttaisi suurelta materiaalilta, joka palaisi lähellä planeetan ilmapiiriä (Kuva 27).

Loin halutun mallin Blenderillä, mille asetin yhden moottoriliekkin tekstuureista.

Osumaan käytin yhtä paineaaltoa ja Kuparisen luomaa puolipalloa. Osuma tehostetta käytettiin myöhemmin EMP hyökkäyksenä, joka lamauttaa lähellä olevat vastustajat.

Loin tehosteesta version, johon lisäsin sähköä ja nopeamman paineaallon.

#### 4.5 Vihollistehosteet



**Kuva 28: Vihollisammuksien sprite sheet ennen ja jälkeen.**

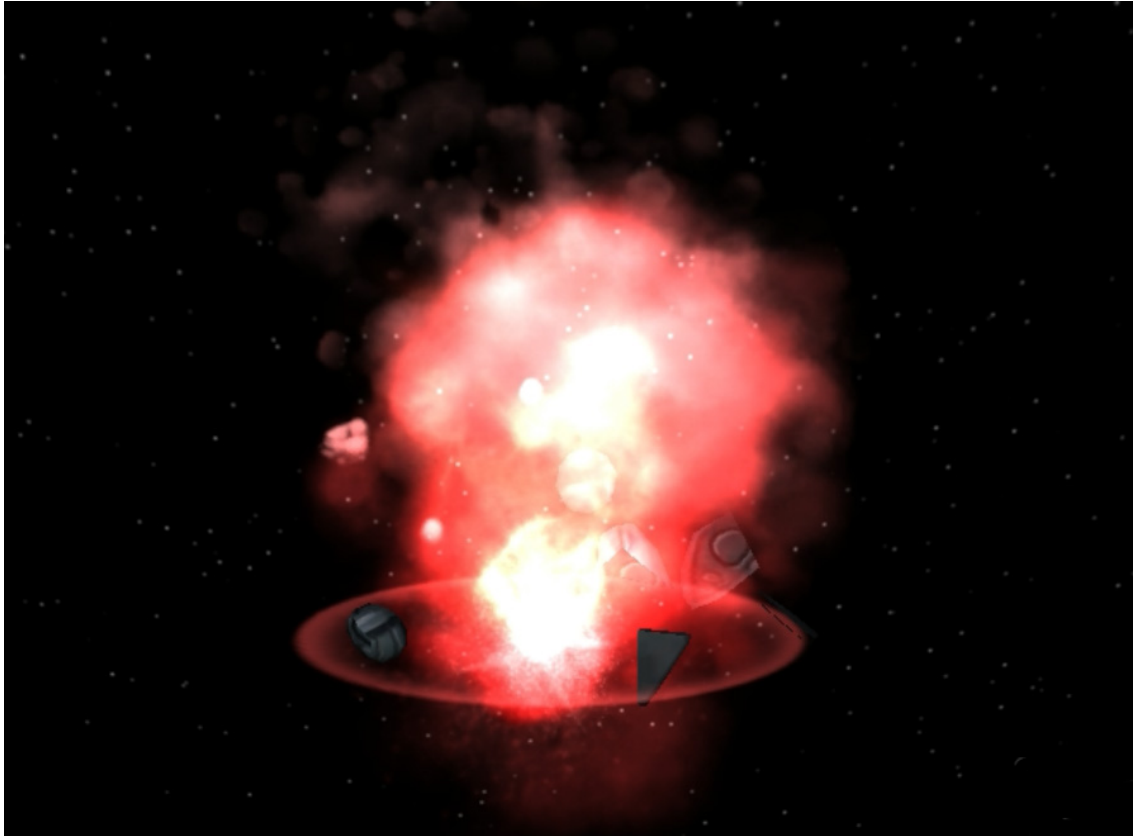
Pelin vihollisina toimivien avaruusolioötköiden tehosteet olivat pitkälti tehtyjä. Kuparinen oli jo tehnyt tekstuurikartat, mutta halusi minun päivittävän niistä paremman näköisiä (Kuva 28). Vihollisammuksia ei animoitu, sillä niitä oli kentällä useita kymmeniä kerralla. Jos ammuksset olisivat animoituja, ne veisivät liikaa laitteen suorituskykyä. Piirsin kartan sykkivän pallon tilalle kolme eri panosta ja päivitin palloammuksen. Tein myös pieniä graafisia päivityksiä loppuihin ammuksiin ja tehosteisiin. Tämän kartan pystyi tekemään värillisesti, sillä kaikki ötköiden ammuksset olivat punaisia. Pelaaja erottaisi ne nopeasti ja helposti, koska punainen kertoo vaarasta (Take Initiative 2014).

Ainoat animoitavat tehosteet olivat neljän framin kestoiset lasersäteet ja viimeisen rivin laserosumatehosteen pohja. Kuparisen toiveesta en tehnyt laserille paljon mitään pienen liikkeen ja hehkun luomisen lisäksi. Animoin liikkeen neljään frameiin jotka olivat aseteltu valmiiksi paikoilleen tekstuurikartassa. Osumatehosteelle en tehnyt mitään ihmeempää päivitystä paineaallon ja pienen tomun lentämisen lisäksi.



**Kuva 29: Swarm ammus.**

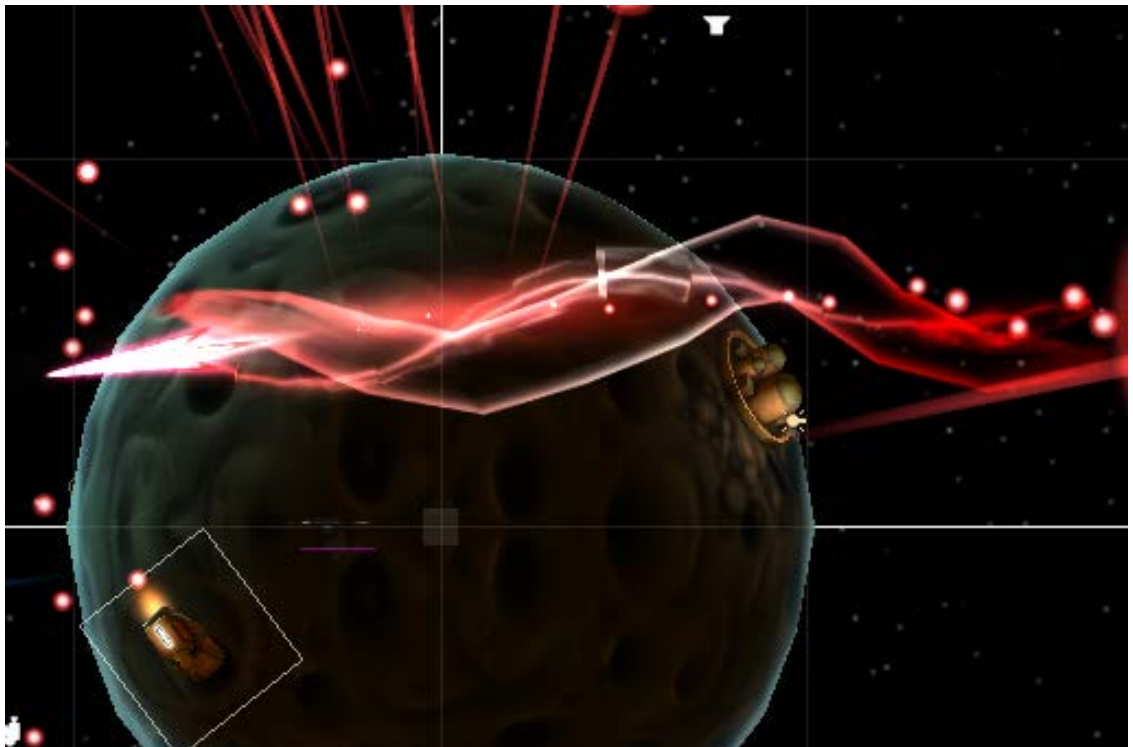
Ainoa joukosta erottuva ammus oli swarm. Se oli kuvattu ampieisparven kaltaisena joukkona pieniä ötököitä, jotka räjähtävät kosketuksesta (Kuva 29). Suunnitelmani mukaan jokaisen parven jäsenen oli pakko pyöriä toistensa ympärillä. Tästä syystä minun oli pakko mallintaa yksi ötökkä, josta tekisin parven. Tein mahdollisimman yksinkertaisen mallin, koska kokonsa vuoksi pelaaja ei saisi selvää parven yksittäisestä jäsenestä, sillä parvi olisi aluksia pienempi. Mallin valmistuttua loin emitterin kopiomaan siitä parven, jotka liikkuvat aavistuksen verran pois päin toisistaan ajan myötä. Viimeinen osuus oli saada parvi pyörimään emitterin akselin ympärillä. Tätä varten jouduin tekemään Emptyn eli tyhjän pisteen emitterin parentiksi. Tälle tyhjälle pisteelle loin animaation, joka pyöri z- ja y-akselin mukaisesti. Ainoa ongelma siinä oli, että parven jäsenet pyörivät holtittomasti animaation mukana, kun niiden olisi kuulunut pysyä koko ajan samansuuntaisina. Kuparinen kertoi, että ongelma oli siinä, että mallit pyörivät oman emitterin akselin mukaisesti. Ratkaisu oli pakottaa partikkelin pivot olemaan aina samansuuntainen maailman pivotin kanssa. Nyt parven jäsenet olivat oikein päin ja pyörivät nädisti toistensa ympärillä.



**Kuva 30: Vihollisen räjähdys.**

Vihollisten kuolinräjähdyksiä tehtiin vain kolme: pieni, keskikokoinen ja suuri. Tällä tavoin minun ei tarvinnut tehdä jokaiselle vihollisalukselle omaa räjähdystä vaan räjähdys määräytyisi ötökän koon mukaan. Tämän lisäksi pystyimme säästämään laitteen muistia. Kierrätin edellisten räjähdysten materiaaleja luodakseni kolme uutta räjähdystä. Isompiin (Kuva 30) lisäsin lentäviä 3D paloja samalla tavalla kuin esimerkiksi satelliitin kuolinräjähdyksen kohdalla.





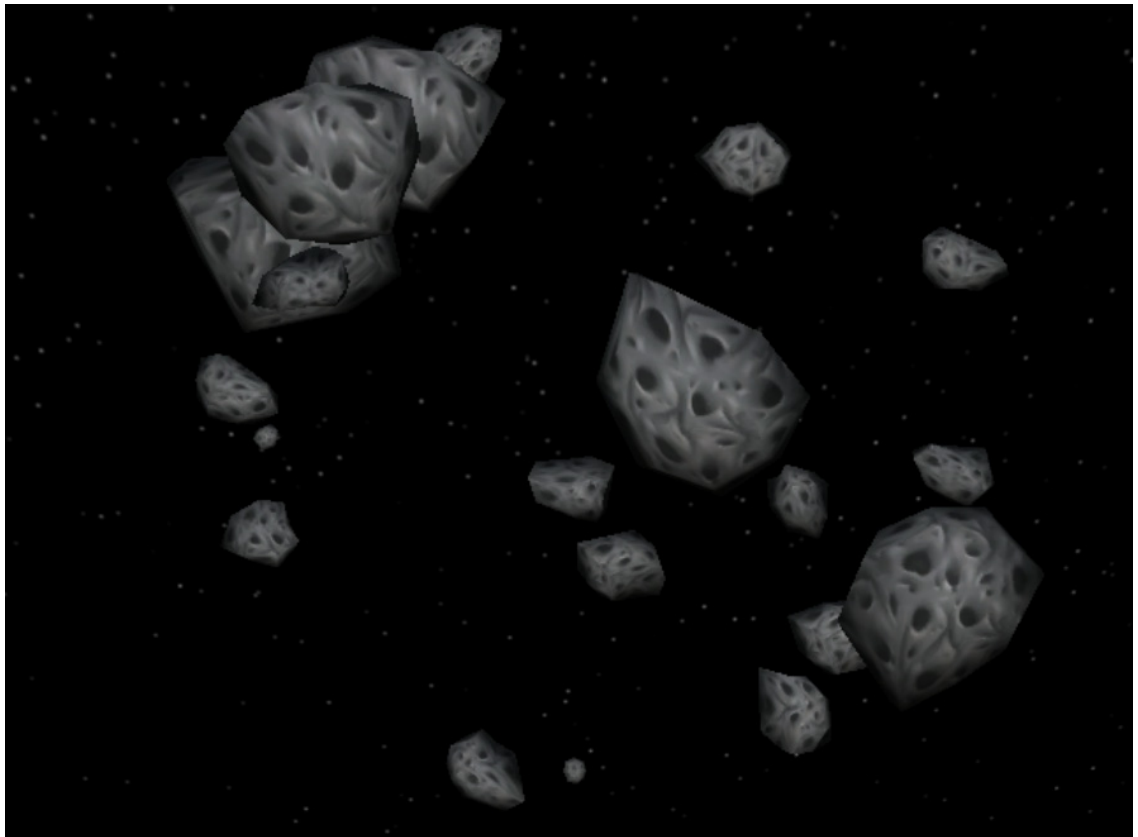
**Kuva 31: Pomoaluksen ammus.**

Viimeinen tehoste vihollisiin liittyen oli pomotaistelussa olevan vihollisaluksen ammus (Kuva 31). Kuparinen oli tehnyt jo ammuksen pohjan animaatioineen. Minun työni oli katsoa, jos sitä voisi parannella hieman ja muutin lähinnä paria materiaalia sekä lisäsin pari emitteriä.

#### 4.6 Ympäristötehosteet

Kuparinen oli lisännyt tehtävälistaani viimeisenä osana ympäristöön kuuluvia erikoistehosteita. Suurin osa oli tehty jo valmiiksi Kuparisen toimesta. Parin uuden tehosteen tekemisen lisäksi hän halusi, että käyn läpi hänen tekemänsä tehosteet ja parantelin niitä oman näkemykseni mukaan tarpeen vaatiessa. Ainoa tehoste, jota paransin oli ilotulitus, joka näkyy pelaajan voittaessa kentän. Lisäykseni oli pieni, mutta merkittävä. Kuollessaan rakettipartikkeli räjähtää rätiseväksi pikku tomuksi. Lisäsin tekstuurin, joka antoi vaikutelman tästä, jotta ilotulitus näyttää realistisemmalta.





**Kuva 32: Asteroiditehoste.**

Ensimmäinen tehoste, joka minun piti luoda alusta loppuun ympäristöjä varten oli asteroideja taustalle (Kuva 32). Kuparinen pyysi minua mallintamaan Blenderillä kolme suurta, kolme keskikokoista ja kolme pientä asteroidia. Hän suunnitteli käyttäviä niitä luodakseen asteroidikentän myöhemmin. Tähän tehosteeseen minun oli myös maalattava saumaton tekstuuri 3D-malleja varten. Tätä tekstuuria voisi myös käyttää myöhemmin planeetan tai kuun pintana. Maalasin tekstuurin käsin Photoshopin perussiveltimellä, hyödyntäen Offset toimintoa saadakseni tekstuurista saumattoman. Korjattuani Offsetin tuomat ongelmat, vein sen Blenderiin aseteltavaksi. Tekstuuri oli hieman vaikea asettaa aluksi laatikon ympärille, sillä tein kartan miten tahansa sauma näkyi jostain kulmasta. Ratkaisuni oli maalata reunat himmeästi saman värisiksi ja saman näköisiksi, mikä toimi tekemissäni malleissa. Mallien teko sujui käden käänteessä muokkaamalla laatikoita kiven järkäleen muotoon. Unityn puolella asetin emitterin renderöimään meteoriitteja. Tässä vaiheessa minulle selvisi, että Unityn yläraja emittereissä esitettävillä malleilla oli neljä eri meshiä. Lisäsimme mallille kuolinanimaation helpottamaan meteoriittien vaihtumista tehosteen elinkaaren lopussa.



**Kuva 33: Musta-aukko ja sprite sheet rivit.**

Yksi nopeasti luotava tehoste ympäristölle oli vihollisen laskeutumisalueen merkitsevä tehoste. Loin sykkivän pallon, johon imeytyy renkaita. Tätä tehostetta tehdessä sain ajatuksen luoda tehoste mustalle aukolle (Kuva 33). Sitä ei ollut työlistalla, mutta halusin kokeilla tehosteen luomista omaksi ilokseni. Aloitin luomalla samankaltaisen pohjan, mikä vihollisen laskeutumistehosteessa oli, ainoastaan isomman ja nopeampiliikkeisen. Olin myös aikaisemmin työni aikana oppinut jotain tätä tehostetta varten kirjoitusvirheen kautta kun sprite sheetin rivejä ilmoitetaan partikkeli-emitterille. Normaalisti sprite sheet animaatiossa ilmoitetaan tarkkaan rivien määrä, jos sprite sheetissä on yksi neljän framen animaatio, ilmoitat että tekstuurissa on kaksi ruutua ylhäältä alas ja kaksi riviä oikealta vasemmalle. Olin vahingossa ilmoittanut, että tekstuurissa on yksi ruutu ylhäältä alas ja kaksi riviä oikealta vasemmalle, minkä emitter ymmärtää kahden framen animaationa (Kuva 32). Tämä johtaa venymiseen tekstuurissa, mitä käytin hyväkseni tehosteessa. Laitoin yhden venyneen tekstuurin pyörimään vasemmalle ja kopion siitä pyörimään oikealle. Tuloksena oli mielenkiintoisen näköinen pyörivä emitteri, joka näytti todella hyvältä tehosteen lisänä.

## 5 ITSEARVIOINTI PROJEKTISTA

Olin todella innostunut projektista, jonka Blackland Games oli antanut opinnäytetyötäni varten. Tämä oli uusi osa-alue minulle ja jouduin aloittamaan aivan alusta ja opettelemaan useampia asioita pelinkehitykseen liittyen. Parasta tässä oli ehdottomasti, että pystyn liittämään tehosteisiin liittyvät työni portfoliooni. Opinnäytetyöni nostatti taitojani alalla ja tunnen oppineeni paljon enemmän tällä työllä kuin vastaavalla, mikä olisi ollut lähempänä omaa mukavuusaluettani. Kyseessä oli täydellinen tasapaino uuden oppimisen ja vanhan kehittämisen välillä. Se, että sain haastavan työn, joka ei ollut minulle mahdoton oppia, mutta vaati aikaa opetella, oli oikea onnenpotku. Blackland Gamesin art director Tero Kuparisen ansiosta olen alkanut tarkkailemaan enemmän liikettä ympäristössä, jossa elän, mikä on kannustanut minua opettelemaan ja hiomaan animaatiotaitojani.

Vaikkakin olen mielestäni saanut aikaan hyviä tehosteita mobiilipeliä varten, suurin ongelmani tämän työn kanssa on, että se oli kirjaimellisesti ensimmäinen kerta kun tein niitä. Osa tehosteista jäi omasta mielestäni hieman keskeneräisiksi ja viimeistely olisi mahdollisesti ollut tarpeen. Ymmärrän kyllä, että animaatiota voi hioa loputtomiin, jotta siitä saadaan paljon joustavampaa ja täydellisesti ajoitettua. Joskus on jatkettava muihin osa-alueisiin, joita tarvitsee tehdä. Silti jokaista tehostetta olisi voinut parantaa. Olisin voinut myös parantaa ja suunnitella sprite sheetejäni. Osassa niistä tein virheitä, joita mainitsin teoriaosuudessa kuten tilankäyttö tekstuurikartoissa. Hyvä esimerkki on osumatehoste (Kuva 16), jonka olin animoinut kasvamaan sprite sheetissä, vaikka sen kasvaminen saadaan tapahtumaan partikkeli-emitterissä. Pieni työstö sieltä täältä olisi ollut hyväksi. Sen lisäksi se olisi tuonut enemmän dynamiikkaa tehosteisiin. Olen myös parantanut animaatiotaitojani opinnäytetyöni loputtua. Jos haluaisin muuttaa jotain, se olisi animaation ajoitus ja sprite sheetin laatu. Olisin voinut mahdollisesti luoda ehkä toisen 16 framin animaation räjähdyksiä varten luomaan enemmän variaatiota.

Optimointi oli toinen asia, jota pysähdyin miettimään enemmän, varsinkin kun opinnäytetyön aikana huomasin kuinka samankaltaisia osa tehosteista on. Esimerkiksi edellä mainitut savu, höyry ja pöly. En ole varma, toimisiko tämä ajatus kuinka hyvin, mutta hyödyntämällä Unityn partikkeli-emitteristä löytyvää sub-emitter toimintoa voisin laittaa emittereitä emitterien sisään. Voisin periaatteessa luoda useamman

tehosteen vähemmällä määrällä spritejä. Luomalla yhden neljän framen animaatiosta koostuvan sprite sheetin savulle, höyrylle ja pölylle voisin luoda jokaiselle kolmelle oman kuolinanimaation neljällä tai useammalla framilla. Toisin sanoen sen sijaan, että loisin yhden 16 framin sprite sheetin voisinkin luoda yhden spite sheetin, jota käytettäisiin pohjana ja useamman eri kuolinanimaatio sprite sheetin, joka liitettäisiin pohjatehosteeseen riippuen siitä minkä tehosteen haluan luoda. Tällä tavoin tehosteissa olisi enemmän vaihtelua, jotta tehosteet eivät näyttäisi liian samankaltaisilta.

Havaintoni erikoistehosteiden luomisesta on todella laajentanut näkemystäni, miten työn tekoa voi lähestyä monesta eri kulmasta. Se on kirjaimellisesti pakottanut minut ajattelemaan entistä monipuolisemmin. Yksikään tapa ei ole väärä, mutta jokainen vaihtoehto avaa eri tapoja lähestyä työtä. Opinnäytetyössäni lähestyin tehosteita yleisimmällä tavalla sen sijaan, että olisin luonut oman tapani. Uskon syyn olevan kokemattomuuteni tällä osa-alueella, mutta siksi se onkin tärkeää. Olen oppinut näkemään ongelmat useammasta kulmasta, mikä varmasti auttaa minua jatkossa. Jos joskus tulevaisuudessa pääsen luomaan lisää tehosteita, osaan varmasti suunnitella ne projektista riippuen useammalla eri toteutustekniikalla.

Opin paljon uutta vanhoista tutuista ohjelmista, sekä kokonaan uusia ohjelmia. Blender on ohjelma, joka oli minulle aivan uusi tuttavuus. Olen edelleen aika ylpeä siitä, kuinka nopeasti opin tämän uuden ohjelman jo työskentelyni alkuvaiheilla. Koska mallien tekstuurit maalattiin suoraan Blenderissä, opin myös perusteet käsinmaalatuille tekstuureille, jotka kuuluivat pelin taidesuuntaukseen. Tekstuurin maalaaminen on todella hyödyllinen taito, joka nosti myös digitaalisia maalaustaitojani. Opin myös uusia tapoja Photoshopissa, joita en ollut ajatellutkaan vaihtoehtoina. Näistä tavoista kaikkein hyödyllisin oli ehdottomasti filttereiden käyttö sprite sheettien luonnissa. En ollut ikinä ennen ajatellut tätä tavaksi luoda tekstuureja spriteille.

Tämä opinnäytetyö on ollut minulle kyllä arvoinen kokemus, ja olen todella kiitollinen siitä mahdollisuudesta, joka minulle annettiin oppia ja kehittyä alalla, josta minulla on vielä paljon opittavaa.

## LÄHTEET

Ahearn, L. 2009. 3D Game Textures: Create Professional Game Art Using Photoshop, 2. painos 2009. Burlington: Focal Press.

Blackman, S. 2013. Beginning 3D Game Development with Unity 4, 2. painos 2013. New York: Apress.

Crytek. VFX Games - Particle Effects. Saatavissa:  
<http://wiki.polycount.com/CategorySpecialEffects> [viitattu 30.1.2014].

Gamedev.net. Make a Particle Explosion Effect. Saatavissa:  
[http://www.gamedev.net/page/resources/\\_/creative/visual-arts/make-a-particle-explosion-effect-r2701](http://www.gamedev.net/page/resources/_/creative/visual-arts/make-a-particle-explosion-effect-r2701) [viitattu 15.12.2013].

Gilland, J. 2009. Elemental Magic: The Art of Special Effects Animation, 2. painos 2013. Oxon: Focal Press.

Gilland, J. 2012. Elemental Magic Volume 2: The Art of Special Effects Animation. Oxford: Focal Press.

Goldstone, W. 2009. Unity Game Development Essentials. Birmingham: Packt Publishing Ltd.

Film reference. Special effects. Saatavissa:  
<http://www.filmreference.com/encyclopedia/Romantic-Comedy-Yugoslavia/Special-Effects-OPTICAL-EFFECTS.html> [viitattu 18.2.2014].

InCreator[EST]. Foorumi viesti. YOYOGAMES. Saatavissa:  
<http://gmc.yoyogames.com/index.php?showtopic=501630> [viitattu 4.2.2014].

KatsBits. Making better textures for games, 'power of two' and proper image dimensions. Saatavissa: <http://www.katsbits.com/tutorials/textures/make-better-textures-correct-size-and-power-of-two.php> [viitattu 8.2.2014].

Kuparinen, T. Suullinen työnohjaus 1.8.2013-16.10.2013. Kouvola: Blackland Games.

Photoshop tutorials for webdesigners. Create a fireball or Mars like planet. Saatavissa:  
<http://www.funphotoart.com/create-a-fireball-or-mars-like-planet> [viitattu 30.1.2014].

Princeton University. Computer-generated Imagery. Saatavissa:  
[http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Computer-generated\\_imagery.html](http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Computer-generated_imagery.html) [viitattu 18.2.2014].

ShadowGun: Optimizing for Mobile Sample Level. Unity - Game Engine. Saatavissa:  
<http://blogs.unity3d.com/2012/03/23/shadowgun-optimizing-for-mobile-sample-level/>  
 [viitattu 15.12.2013].

Take Initiative. Colour within Game Design: Colours with Meaning. Saatavissa:  
<http://www.takeinitiative.co.uk/?p=863> [viitattu 30.1.2014].

Unity 3D. Particles. Saatavissa: <http://wiki.etc.cmu.edu/unity3d/index.php/Particles>  
 [viitattu 30.1.2014].

Unity - Game Engine. Unity Documentation. Saatavissa:  
<http://unity3d.com/learn/documentation> [viitattu 15.12.2013].

Westminster. Special Effects. Saatavissa:  
<https://film110.pbworks.com/w/page/12610294/Special%20Effects> [viitattu  
 18.2.2014].

Williams, R. 2001. The Animator's Survival Kit. Lontoo: Faber and Faber Limited.

Youtube. Animation Photoshop Techniques Electricity. Saatavissa:  
<http://www.youtube.com/watch?v=wfm2mFZ7RLk&list=FLQddOTuXgs93BmtlEdZnPlw&index=39> [viitattu 30.1.2014].